Events Detection and Cyber-Attacks Prediction of Power Grid System Using Machine Learning

By

Md. Muhibbin Hossin Sarder

Roll: 1707081

Department of Computer Science and Engineering

Khulna University of Engineering & Technology

Khulna 9203, Bangladesh

February 2023

Events Detection and Cyber-Attacks Prediction of Power Grid System Using Machine Learning

By

Md. Muhibbin Hossin Sarder

Roll: 1707081

A thesis submitted in partial fulfillment of the requirements for the degree of

"Bachelor of Science in Computer Science & Engineering"

Supervisor:

Signature

Dr. Kazi Md. Rokibul Alam

Professor

Department of Computer Science and Engineering

Khulna University of Engineering Technology

Khulna, Bangladesh.

Acknowledgments

With immense pleasure, I, Md. Muhibbin Hossin Sarder present my thesis work entitled "Events Detection and Cyber-Attacks Prediction of Power Grid System Using Machine Learning" for the partial fulfilment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering in the Department of Computer Science and Engineering, Khulna University of Engineering Technology, Khulna.

I would like to thank the Almighty Allah for His unending blessings on me whenever I need. I express our profound thanks to my Supervisor Dr. Kazi Md. Rokibul Alam, Professor, Dept. of Computer Science and Engineering (CSE), Khulna University of Engineering Technology (KUET) who has directly guided me, encouraging us and helping us throughout the course of this study.

I am also very thankful to all the faculty members of the Dept. of KUET to give me inspiration, suggestions throughout all the courses of me B.Sc. degree which really help me in my thesis study.

February 2021 ***Md Muhibbin Hossin Sarder***

Abstract

For managing energy supply and consumption, the power grids are being digitalized. Natural and man-made event can make disturbance to this digital system. An attack detection model for power system based on machine learning that can be trained by using information and logs collected by phasor measurement units (PMUs). A PMU measures the phasor values of current and voltage. These values get a higher position timestamp and together with the values of power frequency, power frequency change rate and optimal binary data that are also time-stamped are transmitted to a central analysis station. The evaluation of the proposed method is carried out on a benchmark dataset of labeled PMU data, and the results are compared with the different machine learning approaches. The results demonstrate that Random Forest performs better in terms of accuracy, precision, recall, and F1-score.

Contents

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

Introduction

1.1 Background

The power grid is a crucial component of modern society that provides electricity to homes and businesses. However, the reliance on technology and interconnected systems in the power industry has made it vulnerable to cyber attacks. These attacks can have severe consequences, including widespread power outages and economic disruption.

Industrial Control Systems (ICS) frequently use a Cyber-Physical System (CPS) to link the physical and digital worlds so that users can get all necessary information instantly. The application of CPS has great potential in power distribution grids. But unlike conventional cyber security issues, CPS security concerns entail confidentiality, integrity, and availability.

Cyber attacks on the power grid can take many forms, such as denial of service attacks, malware infections, and network breaches. These attacks can destabilize the power grid, leading to widespread power outages and economic disruption.

To address this issue, power grid operators are investing in event and cyber attack detection systems. These systems use advanced algorithms and machine learning techniques to monitor the power grid for unusual activities and to identify potential cyber attacks. By detecting cyber attacks in real-time, power grid operators can quickly take action to mitigate the impact of the attack and protect the power grid from further damage.

In conclusion, the power grid is vulnerable to cyber attacks due to the reliance on technology and interconnected systems in the power industry. The development of event and cyber attack detection systems is crucial to mitigate the threat of these attacks and ensure the stability and reliability of the power grid in the face of growing cyber security threats.

1.2 Motivation

The motivation for event and cyber attack detection in the power grid arises from the need to protect this critical infrastructure from the growing threat of cyber attacks. As the power industry continues to adopt advanced control systems, such as supervisory control and data acquisition (SCADA) systems, the risk of cyber attacks increases. These systems are vulnerable to a variety of cyber attack methods, including denial of service attacks, malware infections, and network breaches.

Unlike conventional cyber security issues, CPS security concerns entail confidentiality, integrity, and availability. Hackers, however may take advantage of flaws to purposefully create branch overload tripping, which might lead to a catastrophic failure and cause significant harm to smart grids.**1** On December 23, 2015, the power grid of Ukraine was hacked, which resulted in power outages for roughly 230,000 consumers in Ukraine for 1-6 hours. The attack took place during an ongoing Russo-Ukrainian War and is attributed to a Russian advanced persistent threat group known as ”Sandworm”.[1]

To mitigate the risk of cyber attacks, power grid operators must have the capability to detect and respond to these attacks in real-time. This requires the development of event and cyber attack detection systems that can monitor the power grid for unusual activities and identify potential cyber attacks.

The development of these systems is crucial for ensuring the stability and reliability of the power grid. By detecting cyber attacks in real-time, power grid operators can take swift action to mitigate the impact of the attack and protect the power grid from further damage. This can help to minimize the consequences of cyber attacks and prevent widespread power outages and economic disruption.

In conclusion, the motivation for event and cyber attack detection in the power grid arises from the need to protect this critical infrastructure from the growing threat of cyber attacks. The development of event and cyber attack detection systems is crucial for ensuring the stability and reliability of the power grid and minimizing the consequences of cyber attacks.

1.3 Scopes of the work

The increasing reliance on technology and interconnected systems in the power industry has made the power grid vulnerable to cyber attacks. These attacks can result in significant consequences, such as widespread power outages and economic disruption. Despite the growing threat of cyber attacks on the power grid, there remains a significant challenge in effectively detecting and responding to these attacks in real-time.

The current methods for detecting cyber attacks on the power grid are often reactive and not proactive, making it difficult to prevent or mitigate the impact of these attacks. This is due to a lack of sophisticated and reliable event and cyber attack detection systems that can effectively monitor the power grid for suspicious activities and identify potential cyber attacks.

Despite the growing threat of cyber attacks on the power grid, there remains a significant challenge in detecting these attacks in a timely and accurate manner. The vast amounts of data generated by the power grid, combined with the dynamic and complex nature of cyber attacks, make it difficult for traditional security systems to effectively detect and respond to these attacks.

Therefore, the problem statement for this thesis is to develop an event and cyber attack detection system for the power grid using machine learning techniques. The goal of this system is to accurately and quickly detect cyber attacks on the power grid, thereby reducing the impact of these attacks and ensuring the stability and reliability of the power grid. This thesis aims to address the challenges of detecting cyber attacks on the power grid by leveraging the power of machine learning algorithms to analyze the vast amounts of data generated by the power grid and identify unusual patterns that may indicate a attack.

1.4 Objectives

The power grid is a critical infrastructure that provides electricity to homes and businesses, and the increasing reliance on technology and interconnected systems has made it vulnerable to cyber attacks. Despite the growing threat of cyber attacks on the power grid, current

event and cyber attack detection methods are insufficient in detecting attacks in real-time and accurately identifying their origin. This results in delayed response times and the potential for widespread damage to the power grid.

The problem is that traditional methods of event and cyber attack detection in the power grid are based on rule-based algorithms that are limited in their ability to detect complex and evolving cyber threats. The increasing volume and complexity of data generated by the power grid also makes it difficult to detect attacks using traditional methods.

To address this problem, a novel approach is needed that leverages the power of machine learning to detect events and cyber attacks in the power grid. Machine learning algorithms have the potential to learn from vast amounts of data, identify patterns, and make predictions in real-time. This makes them well-suited for the task of detecting cyber attacks in the power grid, where large amounts of data are generated and the threat of cyber attacks is constantly evolving.

1.5 Contribution

In order to identify system events, this research develops a machine learning model that primarily learns from historical data and PMU(Phasor Measurement Unit) log information. While training the data with different machine learning models, unsupervised learning offers an advantage in identifying zero-day attacks, but it can also produce a lot of false positives. Additionally, supervised learning can undoubtedly boost the detection's level of confidence. Based on this, we conduct experiments using supervised machine learning techniques. We also used different ensemble learning to get the best result. The following innovative contributions are made by this paper:

i. The dataset used in the thesis contain 37 events in power grid system with 128 features or the PMU values. Firstly we removed the highly correlated features from the dataset. And performing principal component analysis, we reduced the features to 44 which approximately contains about 98% of the main features.

ii. We suggest a approach to deal with unusual data in the datasets, such as Not a Number(NaN) and Infinity values. The proposed strategy can further increase accuracy when compared to the conventional approaches to handling unusual data.

iii. We build a machine learning-based classification model. Our model can successfully distinguish 37 types of behaviors, including power system faults, trip command injection attacks, relay setting change attacks, and single-line-to-ground (SLG) fault replay

attacks, as shown by the average accuracy, precision, recall, F1 score and elapsed time on 15 datasets, which are respectively 0.94053, 0.8950, 0.94053, 0.9404 and 3.665.

1.6 Thesis Organization

The rest of the thesis report is organized as below:

1. Chapter 2: Description about the related works been done on cyber-attack detection in system.
2. Chapter 3: Description about the tools used to complete this study.
3. Chapter 4: Description of the proposed methodology of the study.
4. Chapter 5: Description about the works been done and results on cyber-attack detection.
5. Chapter 6: Conclusion and possible future works.

CHAPTER II

Literature Review

2.1 Introduction

There are several different types of cyberattacks that can target the power grid, including denial-of-service assaults, malware infections, and network intrusions. These assaults have the potential to damage the electrical grid, resulting in widespread power outages and a drop in the economy. Power grid operators are investing in event and cyber attack detection systems to address this problem. These devices monitor the electrical grid for odd activity and spot potential cyberattacks using cutting-edge algorithms and machine learning techniques.

2.2 Related works

PMU data are used in conventional methods to estimate the state of the power system, and the difference between observed and estimated readings is compared with a threshold for cyber attack detection [2][3]. A simple method was put forth in [4], which investigates the spatial-temporal correlations between state estimates of the grid and uses trust-voting to quickly identify aberrant state estimates in smart grids brought on by FDI attacks. For the purpose of identifying cyberattacks on smart grids, the chi-square detector and cosine similarity matching techniques were examined in [5]. For the real-time detection of FDI attacks in smart grids, Huang et al. [6] devised an adaptive cumulative sum (CUSUM) technique.

Machine learning has been very popular recently for identifying cyber-attacks in smart grids. Network systems are frequently used to connect the systems, also known as critical infrastructure systems, in order to monitor and gather data on equipment operations in real time. Supervisory control and data acquisition (SCADA) systems are very vulnerable to cyberattacks, thus these attacks must be managed carefully [7][8]. In smart grids, where the bulk

of suggested approaches are based on supervised learning algorithms, machine learning has recently been employed extensively for identifying cyber intrusions. [9] investigated a number of supervised learning techniques for differentiating between cyberattacks and power system problems. For the purpose of anticipating FDI assaults, Ozay et al. [10] combined ensemble learning and feature-level fusion with a number of well-known supervised algorithms, including perceptron, k-nearest neighbor (KNN), support vector machines (SVMs), and sparse logistic regression (SLR). Their experimental findings show that state estimation-based methods fall short of machine learning techniques in terms of performance.

SVM, KNN, and extended nearest neighbor (ENN) were evaluated by Yan et al. [11] for their effectiveness in identifying both direct and covert FDI attacks on smart grids. Using PMU readings, Singh et al. [12] suggested a decision tree-based anomaly detection method to separate legitimate tripping from malicious attacks and power line problems that trip physical relays. [13] created an Adaboost-based classification model for identifying cyberattacks and power system disturbances utilizing individual PMU data and the random forest as the base classifier. They used feature construction engineering to extract new features from PMU measurements, and for the final detection they coupled classification models with weight voting.

The literature has looked into feature engineering approaches like feature selection and feature extraction in order to increase detection performance and decrease computational complexity.To enhance the effectiveness of supervised learning algorithms on identifying FDI assaults in smart grids, Sakhnini et al. [14] studied three heuristic feature selection approaches, including genetic algorithm (GA), binary cuckoo search (BCS), and particle swarm optimization (PSO). Principal component analysis (PCA), a common feature extraction technique, was employed in [15][16] to minimize the dimensionality of the feature space for a lower computing cost of attack detection.

Ahmed et al. [17] used sensor and process noise fingerprints to detect stealthy cyber-attacks in CPS, and validated the method on a dataset from a real-world water treatment facility, with results indicating an accuracy of up to 98 percent.Shoukry et al.[18] proposed a Multi-Modal

Luenberger observer that can isolate the attacked sensors while estimating the state of the underlying dynamics from the remaining sensors, and their approach is applicable to large-scale CPSs. Hadžiosmanovi ´c et al.[19]developed a semantic, network-based instruction detec- tion system to detect attacks on process control by using network traffic from water plants. Junejo and Goh[20] proposed a behavior-based machine learning approach for the intrusion detection, the dataset they used was SWaT-generated data from 18 attacks of ten different types.

The datasets utilized in this work have also been the subject of several studies. A sequential pattern mining approach was put up by Pan et al. [21] to accurately extract patterns of power system disturbances and cyberattacks. On the 7-class datasets, the mining common path algorithm's accuracy rate was 93%. For the complete multiclass dataset, Hink et al [22] had accuracy rates for JRipper + Adaboost and Random Forest of about 90% and 75%, respectively. In order to pick significant chunks of data and identify intrusive occurrences, Keshk et al. [23] suggested a privacy preservation intrusion detection (PPID) technique based on correlation coefficient and Expectation Maximisation (EM) clustering processes. 88.9% of the multiclass datasets with which the model was tested had recall rates.

The existing reference merely mentioned how to detect attacks in power grids, without delving deeply into the data interaction. Most multi-classification algorithms, on the other hand, transform multi-classification problems into multi-two-class scenarios. Based on the aforementioned consideration, we design a model that performs PCA and then divides the data in accordance with that which was acquired by various PMUs in order to minimize computational overhead. In addition, we use the different machine learning algorithm to find scenarios of cyberattacks and 37-class events in the electrical system.

CHAPTER III

Required Tools

3.1 Data processing and analysis

Numpy, Pandas, Matplotlib for data processing and visualization.

i. **Numpy:** NumPy is a Python library for scientific computing that provides support for arrays and matrices. It provides fast numerical computations and allows operations on arrays and matrices, making it an important tool for data processing and analysis. It also includes functions for linear algebra, random number generation, and basic operations on arrays and matrices.

ii. **Pandas:** Pandas is a data analysis library for Python that provides data structures for efficiently storing and manipulating data. The main data structure provided by Pandas is the DataFrame, which is a two-dimensional labeled data structure that can store a variety of data types. Pandas provides powerful data analysis and data manipulation tools, including indexing, merging, grouping, and reshaping operations.

iii. **Matplotlib:** Matplotlib is a plotting library for Python that provides a variety of plots, including line plots, scatter plots, bar plots, histograms, and others. It provides a high-level interface for producing plots and visualizations, making it easy to create visualizations of data. It also provides customization options, allowing users to control the appearance and behavior of the plots, including color, line width, markers, and others.

3.2 Machine Learning Models

i. **K-Nearest Neighbors:** it's a simple, non-parametric, lazy learning algorithm used for classification and regression problems. It uses a data point's distance to its k nearest neighbors in the training data to make predictions.

   Here's how KNN works:

   (a) Store all the training data points.

   (b) For a new data point to be classified, calculate its distance from all the stored training data points.

   (c) Select the k nearest neighbors based on the distances calculated.

(d) Classify the new data point based on the majority class of the k nearest neighbors.

In KNN, the value of k is an important parameter. A smaller k value means a more complex model and a higher risk of overfitting, while a larger k value means a simpler model and a higher risk of underfitting. The optimal value of k depends on the dataset and the specific problem being solved.

ii. **Decision Tree:** A decision tree is a tree-based algorithm used for classification and regression problems in machine learning. It's a simple and powerful tool that can be used to visualize complex decision processes and make predictions based on input features.

In a decision tree for classification problems, each internal node represents a test on one of the input features, each branch represents an outcome of the test, and each leaf node represents a predicted class label.

Here's how a decision tree for classification works:

(a) Choose the best feature to split the data based on the information gain or reduction in impurity criteria such as entropy or Gini index.

(b) Split the data into subsets based on the values of the chosen feature.

(c) Repeat the process for each subset, creating a new test and branches until a stopping criterion is met (e.g., maximum tree depth, minimum number of samples in a leaf, etc.).

(d) Each leaf node represents a predicted class label, determined by the majority class of the samples in that leaf.

The decision tree algorithm is simple to understand and interpret, but it can suffer from overfitting if the tree is allowed to grow too deep and become too complex. Pruning techniques can be used to reduce the size of the tree and mitigate overfitting.

iii. **Quadratic Discriminant Analysis:** It's a statistical method used for classification problems. It's a more complex and flexible algorithm compared to linear discriminant analysis (LDA), as it allows for non-linear decision boundaries between classes.

In QDA, each class is assumed to have its own covariance matrix, which describes the shape and orientation of the class in the feature space. The decision boundary between classes is then represented by a quadratic surface.

Here's how QDA works for classification problems:

(a) Estimate the mean and covariance matrix for each class based on the training data.

(b) For a new data point, calculate the discriminant function for each class, which is a quadratic expression of the Mahalanobis distance between the data point and the mean of each class.

(c) Classify the new data point based on which class has the largest discriminant function value.

The QDA algorithm is more flexible than LDA but also more computationally intensive and prone to overfitting when the sample size is small compared to the number of features. Regularization techniques can be used to mitigate overfitting in QDA.

iv. **Gradient Boosting:** It's an ensemble machine learning algorithm used for classification and regression problems. It's a boosting algorithm that trains multiple weak models, such as decision trees, in a sequential manner and combines their predictions to form a stronger model.

Here's how GB works for classification problems:

(a) Initialize the prediction for all data points with a constant value, such as the mean of the target variable.

(b) Train a weak model, such as a decision tree, to predict the residuals between the true target values and the current predictions.

(c) Update the predictions by adding the predicted residuals to the current predictions.

(d) Repeat steps 2 and 3 a specified number of times, creating an ensemble of weak models.

(e) Make predictions for new data points by aggregating the predictions of all weak models.

The Gradient Boosting algorithm can be computationally expensive due to the sequential training of weak models, but it can produce highly accurate models when properly tuned. GB algorithms are also highly sensitive to the choice of loss function and regularization parameters, so proper hyperparameter tuning is important for good performance.

v. **eXtreme Gradient Boosting:** it's an optimized implementation of the gradient boosting algorithm used for classification and regression problems. It's an open-source library developed by DMLC that provides an efficient and scalable implementation of gradient boosting for large-scale problems.

In XGB, decision trees are used as the base models and the gradient boosting algorithm is used to optimize the predictions. XGB also provides additional functionality and performance optimizations such as parallel processing, sparsity-aware learning, and weight-based pruning of decision trees.

Here's how XGB works for classification problems:

(a) Initialize the prediction for all data points with a constant value, such as the mean of the target variable.

(b) Train a weak model, such as a decision tree, to predict the residuals between the true target values and the current predictions.

(c) Update the predictions by adding the predicted residuals to the current predictions.

(d) Repeat steps 2 and 3 a specified number of times, creating an ensemble of weak models.

(e) Make predictions for new data points by aggregating the predictions of all weak models.

The XGB library provides an optimized implementation of gradient boosting that can handle large datasets and produce accurate models in a relatively short amount of time. It also provides a convenient interface for model tuning and prediction, making it a popular choice for many practitioners.

vi. **Random Forest:** Random Forest is an ensemble machine learning algorithm used for both classification and regression problems. It's a collection of decision trees, where each tree is trained on a random subset of the data and features. The final prediction is made by aggregating the predictions of all trees, typically through a majority vote for classification problems or a mean prediction for regression problems.

Here's how Random Forest works:

(a) Select a random subset of the data and features to train each decision tree.

(b) Train a decision tree on each subset and make predictions for each data point.

(c) Repeat steps 1 and 2 a specified number of times to create an ensemble of decision trees.

(d) Make predictions for new data points by aggregating the predictions of all trees, either through a majority vote for classification problems or a mean prediction for regression problems.

Random Forest is a powerful algorithm that can handle high-dimensional data and handle non-linear relationships between features and target variable. It also provides a measure of feature importance, which can be used for feature selection and dimensionality reduction. However, it can be computationally expensive for large datasets and large numbers of trees, and overfitting can occur if the number of trees is too large or the tree depth is too deep.

vii. **AdaBoost:** AdaBoost, short for Adaptive Boosting, is an ensemble machine learning algorithm used for both classification and regression problems. It's a boosting algorithm that trains multiple weak models, such as decision stumps, in a sequential manner and adjusts their weights to focus on misclassified examples.

Here's how AdaBoost works for classification problems:

(a) Initialize the weights for all data points to be equal.

(b) Train a weak model, such as a decision stump, to predict the target variable based on the current weights.

(c) Update the weights for misclassified examples to increase their importance in subsequent models.

(d) Repeat steps 2 and 3 a specified number of times, creating an ensemble of weak models.

(e) Make predictions for new data points by aggregating the predictions of all weak models, weighted by their accuracy.

AdaBoost is a powerful algorithm that can handle non-linear relationships between features and target variable, and can be applied to a wide variety of weak models. It's also relatively simple to implement and can be computationally efficient compared to other ensemble methods. However, it can be sensitive to noisy data and outliers, and overfitting can occur if the number of weak models is too large.

viii. **Stacking:** Stacking, also known as stacked generalization, is an ensemble machine learning technique that trains multiple models to make predictions and combines their predictions through a meta-model. The goal of stacking is to leverage the strengths of different models to produce a more accurate prediction than any individual model.

Here's how stacking works for classification problems:

(a) Train multiple base models on the same training data and make predictions for the validation or test data.

(b) Use the predictions of the base models as input features for a meta-model, which is trained to make the final prediction based on the predictions of the base models.

(c) Make predictions for new data points by passing them through the base models to obtain their predictions and using the meta-model to make the final prediction.

Stacking can produce highly accurate models and can be used with a wide variety of base models, including decision trees, linear models, and neural networks. It can also handle heterogeneous data, where different models may be better suited for different parts of the input space. However, stacking can be computationally expensive and can be sensitive to the choice of base models and meta-model. Proper cross-validation and model selection is important to obtain good results.

ix. **Bagging:** Bagging, short for Bootstrapped Aggregating, is an ensemble machine learning technique used for both classification and regression problems. It trains multiple models on random subsets of the data and aggregates their predictions to produce a final prediction. The goal of bagging is to reduce the variance of the model and increase its stability, while maintaining its accuracy.

Here's how bagging works:

(a) Select a random subset of the data with replacement to train each model.

(b) Train a model on each subset.

(c) Repeat steps 1 and 2 a specified number of times, creating an ensemble of models.

(d) Make predictions for new data points by aggregating the predictions of all models, either through a majority vote for classification problems or a mean prediction for regression problems.

Bagging is a powerful technique that can handle high-dimensional data and non-linear relationships between features and target variable. It also provides a measure of model stability, as the ensemble predictions are typically less variable than the predictions of individual models. However, bagging does not reduce the bias of the model, and overfitting can occur if the model complexity is too high. Additionally, bagging can be computationally expensive for large datasets and large numbers of models.

CHAPTER IV

Proposed Methodology

4.1 Overview of the Proposed Method

The proposed method is illustrated in Figure 4.1 depicts a model structural diagram for identifying events and cyber-attacks in a power grid system.

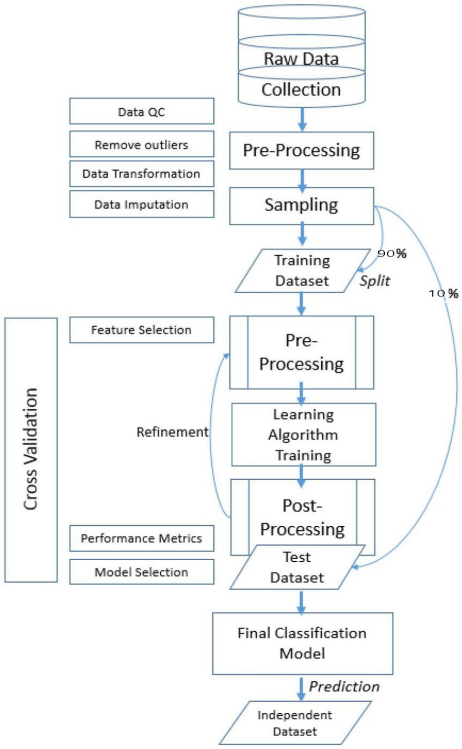


Figure 4.1: Overview of model for detecting disturbance and cyber-attack in power grid

4.2 Data Processing

Data processing is an important step getting a better result, the data processing methods that we used include:

4.2.1 Data Cleaning

Removing missing values, duplicate data, and inconsistent data to ensure that the data is of high quality and suitable for analysis. We removed the instances with missing values, imputing missing values with statistical methods mean and k-Nearest Neighbors algorithm imputation. Here's a more detailed explanation of how k-Nearest Neighbors algorithm imputation works:

1. Calculate the distance between instances: The first step is to calculate the distance between each instance in the dataset. This can be done using any distance metric, such as Euclidean distance, Manhattan distance, or cosine similarity.

2. Find the k nearest neighbors: For each instance with missing values, the k nearest neighbors are then found based on the calculated distances. k is a hyperparameter that determines the number of nearest neighbors to consider.

3. Impute the missing values: The missing values are then imputed based on the values of the k nearest neighbors. This can be done in several ways, such as taking the average of the k nearest neighbors, using a weighted average that gives more weight to closer neighbors, or using a more complex model such as linear regression.

4. Repeat for all instances: Steps 2 and 3 are repeated for all instances with missing values until all missing values have been imputed.

4.2.2 Feature Engineering

Feature engineering is the process of selecting and transforming raw data into useful features that can improve the performance of machine learning models. In other words, it involves creating new features from existing ones, selecting the most relevant features, and transforming them in a way that makes them more suitable for use in a particular machine learning model.

The importance of feature engineering can not be overstated as it can significantly impact the performance of a machine learning model. A good feature engineering process can help to:

i. Improve the accuracy of the model

ii. Reduce overfitting or underfitting

iii. Make the model more interpretable

iv. Reduce the time required for training the model.

Table 4.1: Description of Extracted Features

Feature	Description
VCM1	(PM1:V-PM7:V)/(PM4:I-PM10:I)
VCM2	(PM2:V-PM8:V)/(PM5:I-PM11:I)
VCA1	sin(PA1:VH-:PA4:IH-PA7:VH-PA10:IH)
VCA4	PA7:VH-PA10:IH
sI	sin(PA4:IH-PA10:IH)
sV	sin(PA1:VH-PA7:VH)

When performing feature construction engineering, 16 new fea- tures are extracted in the light of the physical meaning of each PMU measurement feature and then added to the original dataset to prepare for the next step. We mainly adopt relevant calcula- tion measures to extract new features based on raw data. The name, description and extraction method of the extracted feature are shown in Table 4.1 .

4.2.3 Data Transformation

Transforming data into a format that can be easily understood and analyzed. This can include normalizing data, encoding categorical variables, and scaling data.

We used z-score normalization for data processing.The Z-score normalization is a process of transforming a variable into its standard score, which represents how many standard deviations it is from the mean of the variable. The standard score is calculated as follows:

$$z = \frac{x-\mu}{\sigma}$$

where:

z is the Z-score of the variable x is the original value of the variable μ is the mean of the variable σ is the standard deviation of the variable The Z-score normalization helps to scale the variable so that it has a mean of zero and a standard deviation of 1, making it easier to compare the variable to other variables.

4.2.4 Data Reduction

Reducing the size of the data by selecting a subset of the data that is most relevant to the problem being addressed.

Feature Selection: The process of selecting a subset of the features or variables in the data that are most relevant to the problem being addressed. The proposed method is correlation coeffecient and Feature Importance.

Correlation Matrix: A correlation matrix is a table showing the correlation between multiple variables. Each cell in the table contains the Pearson correlation coefficient, which measures the linear relationship between two variables. The Pearson correlation coefficient is defined as:

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y}$$

Where X and Y are two variables, $cov(X,Y)$ is the covariance between X and Y, and σ_X and σ_Y are the standard deviations of X and Y, respectively.

The correlation matrix is a useful tool for understanding the relationships between variables and for identifying multicollinearity, which can affect the stability and interpretability of regression models.

Feature Importance: XGBoost (Extreme Gradient Boosting) is a popular machine learning library that is widely used for classification and regression tasks. One of the key advantages of XGBoost is its ability to provide feature importance scores, which can help users to understand the relative importance of different features in predicting the target variable.

The feature importance scores in XGBoost are calculated based on the number of times each feature is used to split the data across all decision trees in the model. Features that are frequently used for splitting the data are considered more important, while features that are rarely used are considered less important.

There are two types of feature importance scores that XGBoost provides:

1. Weight-based importance: This is calculated based on the number of times a feature

is used to split the data across all trees. Features with higher weights are considered more important.

2. Gain-based importance: This is calculated based on the improvement in the objective function (e.g., mean squared error or log loss) resulting from splitting the data on a particular feature. Features that result in a larger improvement in the objective function are considered more important.

The proposed method is to utilize weight to calculate feature importance

4.3 Establish Supervised Machine Learning Models

i. K-Nearest Neighbors (KNN): KNN is a non-parametric, instance-based, supervised learning algorithm that classifies new data points based on the closest k data points in the training set. The prediction for a new data point is given by:

$$\hat{y} = \frac{1}{k} \sum_{i \in N_k} y_i$$

Where $\hat{y}$ is the prediction for the new data point, N_k is the set of k nearest neighbors, and y_i is the target variable for the i^{th} neighbor.

ii. Decision Trees (DT): DT is a non-linear, non-parametric, supervised learning algorithm that recursively splits the data into smaller subgroups based on the feature that provides the maximum reduction in impurity. The prediction for a new data point is given by:

$$\hat{y} = \text{leaf node value}$$

Where $\hat{y}$ is the prediction for the new data point and the "leaf node value" is the target variable value associated with the leaf node that the new data point falls into.

iii. Quadratic Discriminant Analysis (QDA): QDA is a discriminative, linear, supervised learning algorithm that models the distribution of the target variable given the feature values and class label. The prediction for a new data point is given by:

$$\hat{y} = \text{argmax}_k P(y = k|x)$$

Where $\hat{y}$ is the prediction for the new data point, $P(y = k|x)$ is the posterior probability of the class label k given the feature values x, and argmax_k is the argument that maximizes the posterior probability.

iv. Gradient Boosting (GB): GB is an iterative, non-linear, supervised learning algorithm that trains weak models, such as decision trees, and combines their predictions to form a final prediction. The prediction for a new data point is given by:

$$\hat{y} = \sum_{i=1}^{m} f_i(x)$$

Where $\hat{y}$ is the prediction for the new data point, $f_i(x)$ is the prediction of the i^{th} weak model for the new data point, and m is the number of weak models.

v. XGBoost (XGB): XGB is an optimized version of gradient boosting that uses efficient data structures and parallel processing to improve the speed and accuracy of the model. The prediction for a new data point is the same as gradient boosting, given by:

$$\hat{y} = \sum_{i=1}^{m} f_i(x)$$

vi. Random Forest (RF): RF is an ensemble learning method for classification and regression. It builds multiple decision trees, and the final prediction is made by aggregating the predictions of individual trees (majority voting in classification, average in regression). Each tree is grown on a bootstrapped sample of the training data and a random subset of the features is used to split each node in the tree. The formula for prediction in a random forest can be written as:

$$\hat{y} = \frac{1}{n} \sum_{i=1}^{n} y_i^{tree}$$

Where n is the number of trees in the forest, and y_i^{tree} is the prediction of the i-th tree.

vii. AdaBoost: AdaBoost (Adaptive Boosting) is an ensemble learning method for classification. It builds multiple weak learners in a sequential manner, each trying to correct the mistakes of the previous one. The final prediction is made by weighting the predictions of individual weak learners, with more weight given to the ones that perform better. The formula for prediction in AdaBoost can be written as:

$$\hat{y} = \text{sign}\left(\sum_{i=1}^{n} w_i h_i(x)\right)$$

Where n is the number of weak learners, $h_i(x)$ is the prediction of the i-th weak learner, and w_i is its weight.

viii. Stacking: Stacking is an ensemble learning method for classification and regression. It involves training multiple base models and using their predictions as inputs to a higher-level model (meta-model) to make the final prediction. The formula for prediction in

Stacking can be written as:

$$\hat{y} = \text{Meta-model}(\text{Base-model } 1(x), \text{Base-model } 2(x), ..., \text{Base-model n}(x))$$

Where n is the number of base models, and Base-model i(x) is the prediction of the i-th base model for the input sample x.

ix. Bagging: Bagging (Bootstrap Aggregating) is an ensemble learning method for classification and regression. It builds multiple models on bootstrapped samples of the training data, and the final prediction is made by aggregating the predictions of individual models (majority voting in classification, average in regression). The formula for prediction in Bagging can be written as:

$$\hat{y} = \frac{1}{n} \sum_{i=1}^{n} y_i^{model}$$

Where n is the number of models, and y_i^{model} is the prediction of the i-th model.

CHAPTER V

Prototype Evaluation

5.1 Experimental Setup

5.1.1 Environment

1. **Machine Specification:** The system was run on the following machine configuration. The elapsed time was calculated on this machine which can may differ if using different machine configuration.

   Operating System: Windows 11 Pro N
   Processor: Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz - 2.71 GHz
   Memory: 12.0 GB RAM
   Graphics Card: NVIDIA GeForce 930MX
   Storage: 256 GB SSD

2. **Jupyter Notebook:** Jupyter Notebook has been used as a Python editor. Reason behind using Jupyter Notebook:

   i. **Web-based interface:** Jupyter Notebook provides a web-based interface for creating and editing Python code, making it accessible from anywhere with an internet connection.

   ii. **Support for multiple programming languages:** Jupyter Notebook supports a wide range of programming languages, including Python, making it a useful tool for Python development.

   iii. **Code execution:** Jupyter Notebook allows users to run Python code cells and see the results directly in the browser, making it easy to iteratively develop and test code.

   iv. **Debugging:** Jupyter Notebook provides basic debugging capabilities, such as setting breakpoints and stepping through code, for Python development.

   v. **Data visualization:** Jupyter Notebook supports the creation of visualizations using popular libraries in Python, such as Matplotlib, Seaborn, and Plotly, making it easy to explore and present data.

The UI of Jupyter Notebook shown in Figure 5.1.

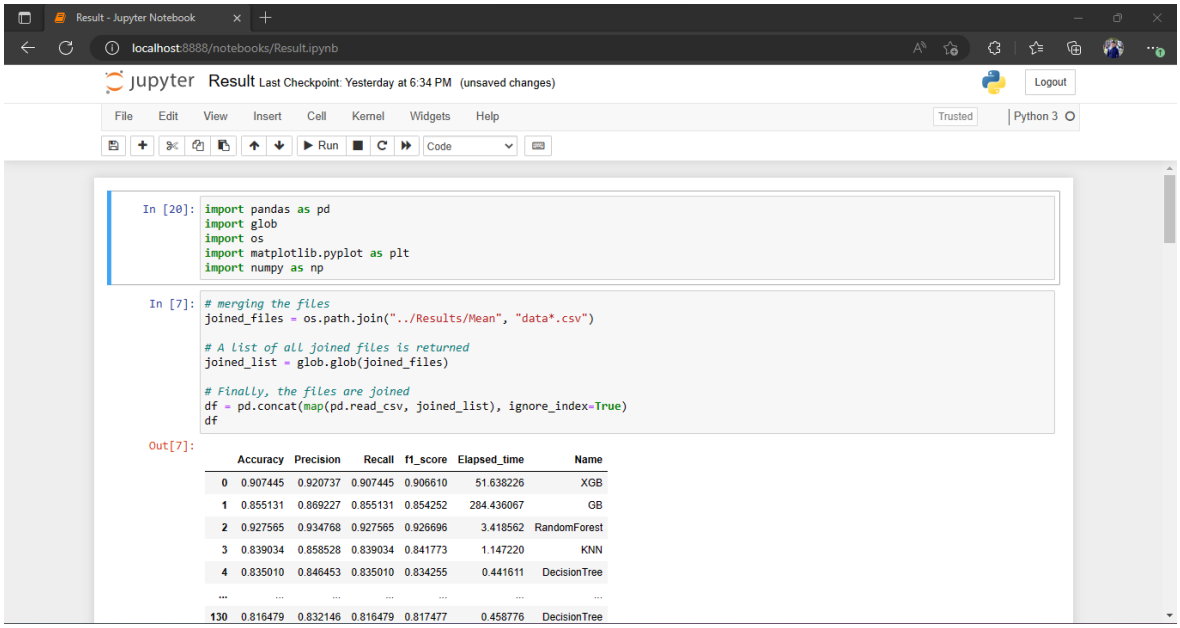


Figure 5.1: UI of Jupyter Notebook

5.1.2 Power System Framework Configuration

The Figure 5.2 shows the power system framework configuration used in generating these scenarios. In the network diagram we have several components, firstly, G1 and G2 are power generators. R1 through R4 are Intelligent Electronic Devices (IEDs) that can switch the breakers on or off. These breakers are labeled BR1 through BR4. We also have two lines. Line One spans from breaker one (BR1) to breaker two (BR2) and Line Two spans from breaker three (BR3) to breaker four (BR4). Each IED automatically controls one breaker. R1 controls BR1, R2 controls BR2 and son on accordingly. The IEDs use a distance protection scheme which trips the breaker on detected faults whether actually valid or faked since they have no internal validation to detect the difference. Operators can also manually issue commands to the IEDs R1 through R4 to manually trip the breakers BR1 through BR4. The manual override is used when performing maintenance on the lines or other system components.

5.1.3 Cyber-Attack Datasets

The datasets used in our study were created from a power system framework [24] made up of smart electronic devices, supervisory control systems, and network monitoring devices, as

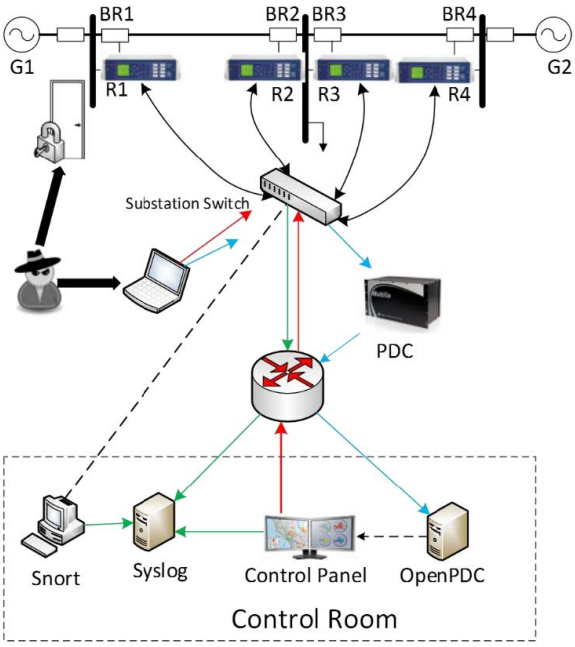


Figure 5.2: The power system framework configuration

shown in Figure 5.2. These datasets are publicly accessible at https://sites.google.com/a/uah.edu/tommy-morris-uah/ics-data-sets (accessed on June 9, 2022). The dataset includes 128 features that collected and recorded by PMUs, relay snort alarm and logs (PMU and relay are integrated).

Table 5.1 shows the situation of each group of dataset. The data distribution is relatively uniform for different label types. The original file format of datasets is ARFF (Attribute-Relation File Format). ARFF file is an ASCII text file that describes a list of in- stances sharing a set of attributes. For convenience, we will convert ARFF files into CSV (Comma Separated Values) format files. CSV file stores numeric or textual tabular data in plain text.

PMU or synchronous phasor is a device that uses a common time source to measure electric waves on a power grid. The 128 features are explained in the Table 5.2. There are 29 types of measurements from each phasor measurement units (PMU). A phasor measurement unit (PMU) or synchrophasor is a device which measures the electrical waves on an electricity grid, using a common time source for synchronization. In our system there are 4 PMUs which measure 29 features for 116 PMU measurement columns total. The index of each

Table 5.1: Multi-class sample data statistics

Dataset	numbers of data
data1	4966
data2	5069
data3	5415
data4	5202
data5	5161
data6	4967
data7	5236
data8	5315
data9	5340
data10	5569
data11	5251
data12	5224
data13	5271
data14	5115
data15	5276
Total	78377

Table 5.2: Description of features measured by a PMU.

Features (No.)	Description
PA1:VH-PA3:VH	Phase A–Phase C Voltage Phase Angle
PM1:V-PM3:V	Phase A–Phase C Voltage Magnitude
PA4:IH-PA6:IH	Phase A–Phase C Current Phase Angle
PM4:I-PM6:I	Phase A–Phase C Current Magnitude
PA7:VH-PA9:VH	Pos.–Neg.–Zero Voltage Phase Angle
PM7:V-PM9:V	Pos.–Neg.–Zero Voltage Magnitude
PA10:VH-PA12:VH	Pos.–Neg.–Zero Current Phase Angle
PM10:V-PM12:V	Pos.–Neg.–Zero Current Magnitude
F	Frequency for relays
DF	Frequency Delta (dF/dt) for relays
PA:Z	Appearance Impedance for relays
PA:ZH	Appearance Impedance Angle for relays
S	Status Flag for relays

Table 5.3: Description of Scenarios.

Scenario No.	Description	Type
41	Normal operation load changes	No Events
1–6	SLG faults	Natural Events
13, 14	Line maintenance	
7–12	Data injection	Intrusion Events
15–20	Remote tripping command injection	
21–30, 35–40	Relay setting change	

column is in the form of "R#-Signal Reference" that indicates a type of measurement from a PMU specified by "R#". The signal references and corresponding descriptions are listed below. For example, R1-PA1:VH means Phase A voltage phase angle measured by PMU R1. After the PMU measurement columns, there are 12 columns for control panel logs, Snort alerts and relay logs of the 4 PMU/relay (relay and PMU are integrated together). The last column is the marker. The first three digits on the right is the load condition (in Megawatt). Another three digits to their left is fault locations, for example, "085" means fault at 85% of the transmission line specified by scenario description. However, for those that do not involve fault, e.g. "line maintenance", these digits will be set to 000. The most left one digit or two digits indicate(s) the scenario number.

The No Events, Natural Events, and Intrusion Events scenario categories make up the majority of the multi-class classification dataset. The scenarios are summarized in Table 5.3

5.2 Experimental Results

We begin by ranking the relevance of the features in order to further investigate how well the features built by the feature construction engineering in the model operate. For model interpretability, feature importance might be used. The importance features are shown in Figure 5.4 As in Figure 5.4, determine the importance of each aspect. The ordinate is the assessment score, and the abscissa is the designation for the significant attributes. The origin features are represented by the gray portion, while the features produced via feature construction engineering are shown by the red mark. All 16 built-in features can be seen to be among the top 40.

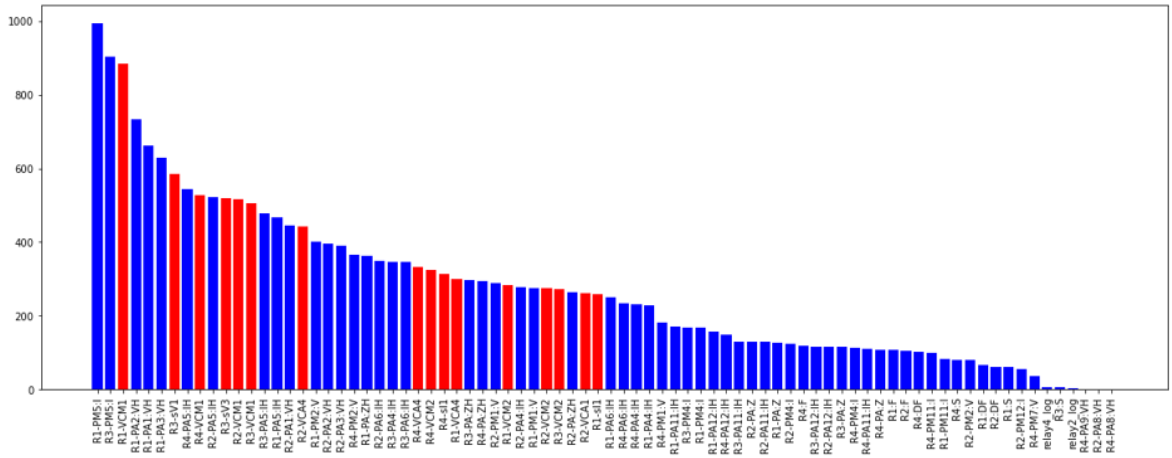


Figure 5.3: Importance features score.

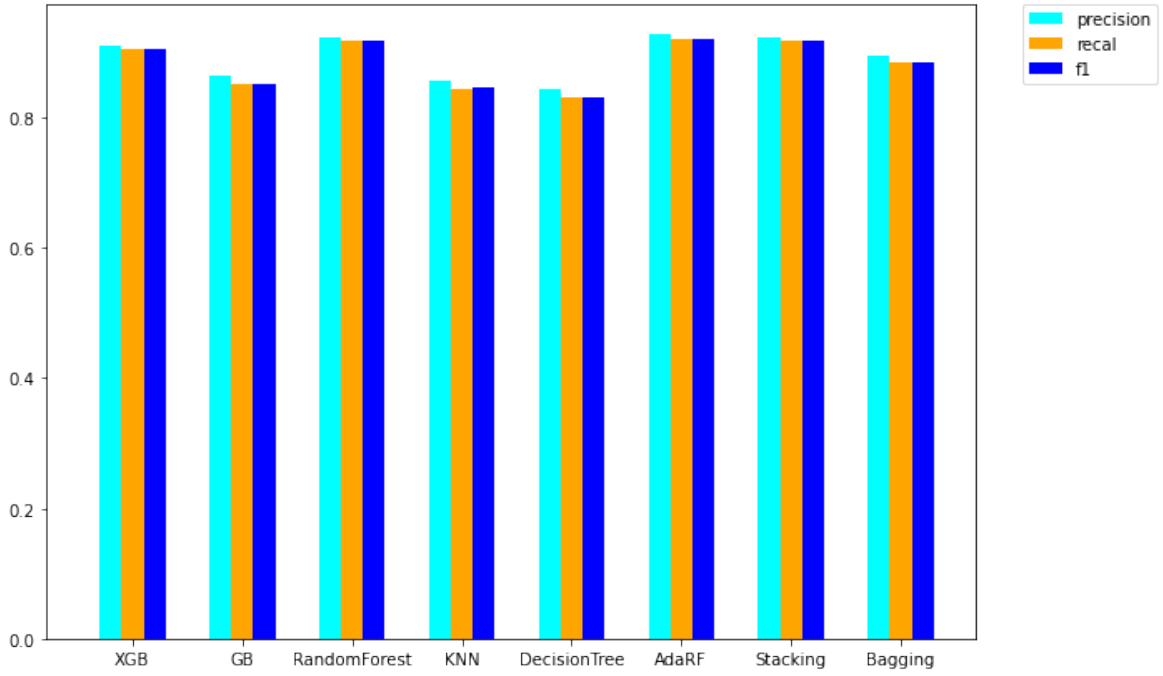


Figure 5.4: precision, recall and f1 score of different model

Table 5.4: Assessment measures of different model

Name	Accuracy	Precision	Recall	f1 score
XGB	0.905143	0.911354	0.905143	0.904522
GB	0.852671	0.863340	0.852671	0.851445
RandomForest	0.917866	0.923823	0.917866	0.917866
KNN	0.848239	0.859720	0.848239	0.848441
DecisionTree	0.830420	0.843068	0.830420	0.830350
AdaRF	0.918500	0.924451	0.918500	0.918386
Stacking	0.915467	0.921791	0.915467	0.914994
Bagging	0.878707	0.889685	0.878707	0.877639

Table 5.5: Assessment measures of using mean and interpolation

Name	Accuracy	Precision	Recall	f1 score
Mean	0.918500	0.0.918500	0.918500	0.918386
Interpolation	0.921432	0.927263	0.921432	0.921432

15 groups of multi-class classification datasets were used in the experiment; each group was trained and evaluated, and the accuracy, precision, recall and f1 score was used as the assessment measure.

The following comparative experiments are carried out using 37 power system event scenarios as the basis for the studies, which serve to demonstrate the need for developing various models based on various PMU types. While the other group sends all features to various machine learning models.

We performed a comparison experiment to confirm the practical use of this strategy. 15 multi-class datasets are used in the experiment, from which we randomly choose the training set and test set in a 9:1 ratio. The 15 training datasets are then combined to form one training set. The model receives this training set for instruction and learning. Finally, 15 test sets are delivered to the model to mimic how it would perform in real-world scenarios.

We used KNN, Decision Tree, XGBOOST, Gradient Boosting, Random Forest, AdaBoost, Stacking and Bagging. By averaging the assessment measures of all the dataset we get the result in Table 5.4. Figure 5.5 shows the accuracy and Figure 5.6 shows the precision, recall and f1 score of the models. After extensive testing, it has been shown that the AdaBoost classifier model performs best on the dataset.

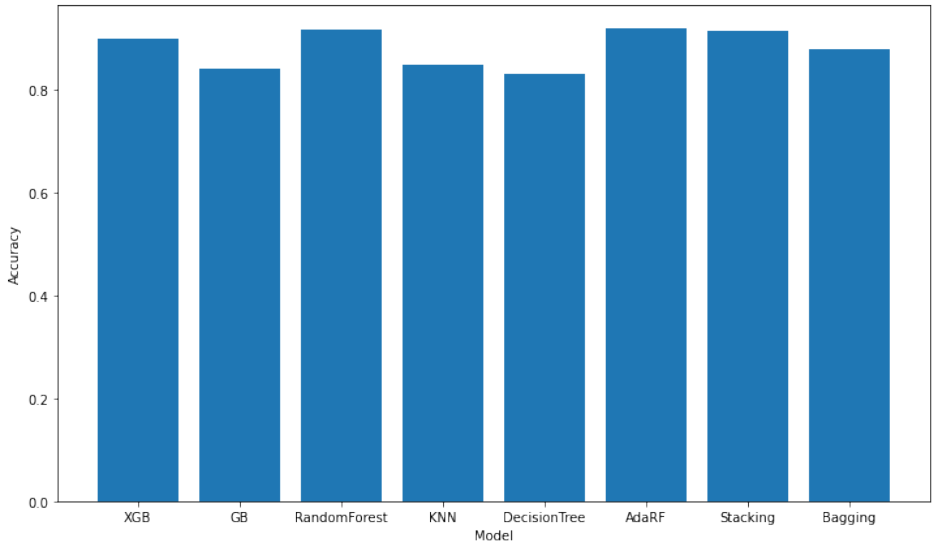


Figure 5.5: Accuracy of different model

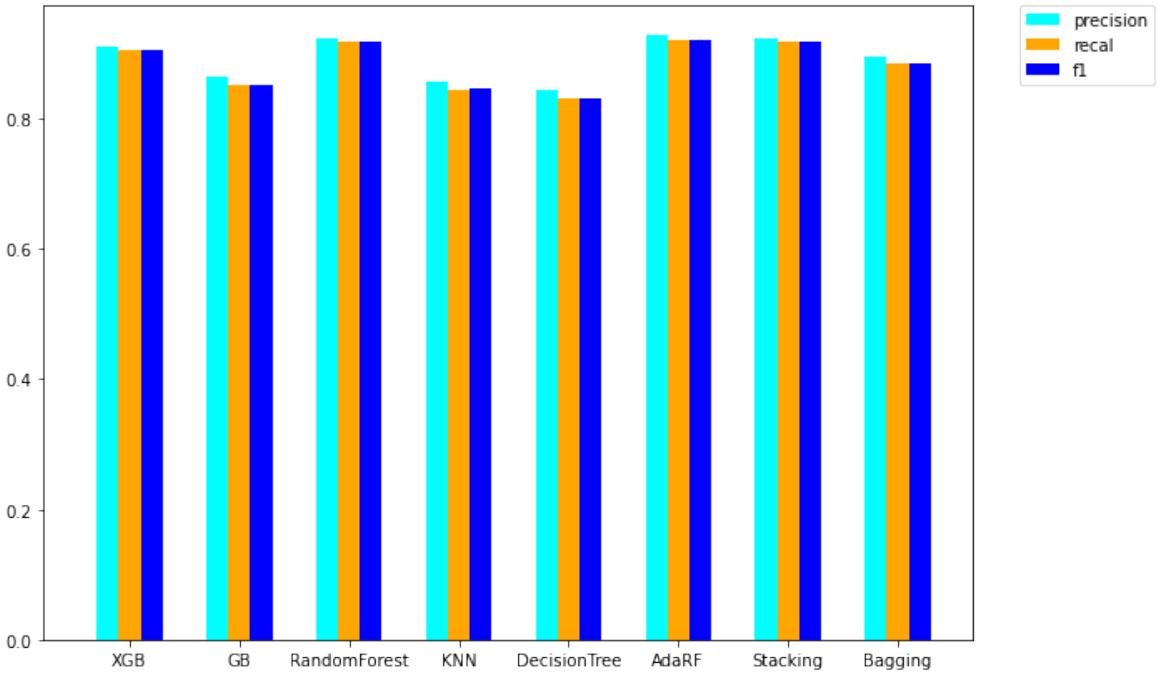


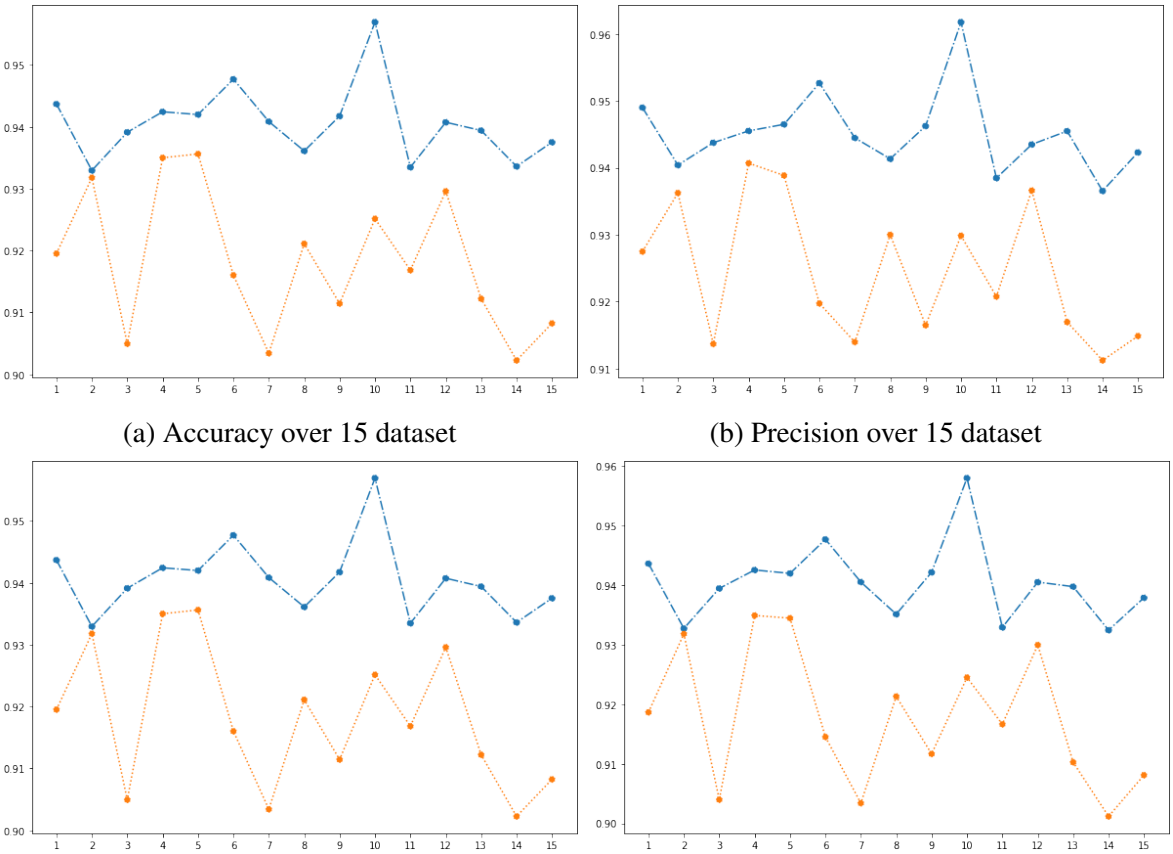
Figure 5.6: precision, recall and f1 score of different model

5.3 Comparisons

We can see from Table 5.4 that Adaboost gives the best accuracy, precision, recall and f1 score. While working with 128 features we got this value of accuracy 0.9182 while after data processing steps the accuracy got upto 0.94053. Also precision, recall and f1 score also increased. Using interpolation as the removal of missing and infinite value also increases the performance showing in Table 5.5. A detailed comparison between the usage of original features and features combined from several assessment measures, such as precision, recall, F1 score, and accuracy by utilizing an AdaBoost model, is shown in Figure 5.7. The ordinate denotes evaluation metrics, whereas the abscissa indicates the dataset's name. It goes without saying that employing combined features for all evaluation criteria is preferable than using original data. It supports the notion that a model will perform better the better its features, as was stated above. In the event of power grid attacks or faults, the proposed model, feature construction engineering, and data processing operations are effective. Compared to the conventional machine learning model, these updates increase the accuracy.

5.4 Discussion

It is visible that the performance increases after combining the feature and data processing rather than taking the full 128 features. We see that the PCA does not perform well in improving the performance of the model. We used Feature selection using XGBoost and reduced the features to 40 by taking the top 40 features and got expected performance from the model. The performance also increased by using the KNN interpolation for replacing the missing and infinite value. The AdaBoost with random forest classifier gives the best performance and another ensemble method stacking gives almost the same performance as well. Obviously, using combined features is better than using original data in all evaluation metrics. Compared to the original features, the dimension of combined features is only 31 dimensions, and the dimension is reduced by 75%, thereby saving computing resources. It verifies the above mentioned that the better the features, the better performance of the model.



(a) Accuracy over 15 dataset

(b) Precision over 15 dataset

(c) Recall over 15 dataset

(d) F1 score over 15 dataset

Figure 5.7: comparison with features using different evaluation matrics

CHAPTER VI

Conclusions

6.1 Conclusions

This study does feature construction engineering for the original data and uses a large number of power grid data as the experimental basis. Additionally, we provide a model for locating power system failures and cyberattacks. The model suggested in this research and conventional machine learning models in the experiment are both evaluated using a number of machine learning evaluation metrics. The findings demonstrate that the data processing technique can enhance the model's experimental accuracy, and the AdaBoost model can successfully identify 37 different power grid behaviors. As a result, it is possible to utilize machine learning techniques in the power system to assist operators in making decisions. The training period will be extended on the current basis due to the growing volume of data. Additionally, it will take a significant amount of resources to manually construct the features after analyzing the precise physical meaning for various dataset types.

6.2 Future Works

Deep learning is essential to Big Data solutions because it can extract useful information from complex systems. We will continue to enhance relevant data and work on deep learning models combined with Big Data processing in the future to address these issues.

The training of supervised algorithms needs both normal and attack data. However, collecting representative instances of various attack events is usually a difficult task if not impossible, which could result in a model with bad performance in detecting certain attacks, especially attack types not represented in training data. We will also work on semisupervised learning in this case in the future.

REFERENCES

[1] D. U. Case, "Analysis of the cyber attack on the ukrainian power grid," *Electricity Information Sharing and Analysis Center (E-ISAC)*, vol. 388, pp. 1–29, 2016.

[2] T. T. Kim and H. V. Poor, "Strategic protection against data injection attacks on power grids," *IEEE Transactions on Smart Grid*, vol. 2, no. 2, pp. 326–333, 2011. DOI: 10.1109/TSG.2011.2119336.

[3] M. Ozay, I. Esnaola, F. T. Y. Vural, S. R. Kulkarni, and H. V. Poor, "Sparse attack construction and state estimation in the smart grid: Centralized and distributed models," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 7, pp. 1306–1318, 2013. DOI: 10.1109/JSAC.2013.130713.

[4] P.-Y. Chen, S. Yang, J. A. McCann, J. Lin, and X. Yang, "Detection of false data injection attacks in smart-grid systems," *IEEE Communications Magazine*, vol. 53, no. 2, pp. 206–213, 2015. DOI: 10.1109/MCOM.2015.7045410.

[5] D. B. Rawat and C. Bajracharya, "Detection of false data injection attacks in smart grid communication systems," *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1652–1656, 2015. DOI: 10.1109/LSP.2015.2421935.

[6] Y. Huang, J. Tang, Y. Cheng, H. Li, K. A. Campbell, and Z. Han, "Real-time detection of false data injection in smart grid networks: An adaptive cusum method and analysis," *IEEE Systems Journal*, vol. 10, no. 2, pp. 532–543, 2016. DOI: 10.1109/JSYST.2014.2323266.

[7] S. Karnouskos, "Stuxnet worm impact on industrial cyber-physical system security," in *IECON 2011 - 37th Annual Conference of the IEEE Industrial Electronics Society*, 2011, pp. 4490–4494. DOI: 10.1109/IECON.2011.6120048.

[8] B. Zhu, A. Joseph, and S. Sastry, "A taxonomy of cyber attacks on scada systems," in *2011 International Conference on Internet of Things and 4th International Conference*

on Cyber, Physical and Social Computing, 2011, pp. 380–388. DOI: 10.1109/iThings/CPSCom.2011.34.

[9] R. Hink, J. Beaver, M. Buckner, T. Morris, U. Adhikari, and S. Pan, "Machine learning for power system disturbance and cyber-attack discrimination," in *Proceedings of the 2014 7th International Symposium on Resilient Control Systems (ISRCS)*, Denver, CO, USA, Aug. 2014, pp. 1–8.

[10] M. Ozay, I. Esnaola, F. T. Yarman Vural, S. R. Kulkarni, and H. V. Poor, "Machine learning methods for attack detection in the smart grid," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 8, pp. 1773–1786, 2016. DOI: 10.1109/TNNLS.2015.2404803.

[11] J. Yan, B. Tang, and H. He, "Detection of false data attacks in smart grid with supervised learning," in *Proceedings of the 2016 International Joint Conference on Neural Networks (IJCNN)*, Vancouver, BC, Canada, Jul. 2016, pp. 1395–1402.

[12] V. Singh and M. Govindarasu, "Decision tree based anomaly detection for remedial action scheme in smart grid using pmu data," in *Proceedings of the 2018 IEEE Power Energy Society General Meeting (PESGM)*, Portland, OR, USA, Aug. 2018, pp. 1–5.

[13] D. Wang, X. Wang, Y. Zhang, and L. Jin, "Detection of power grid disturbances and cyber-attacks based on machine learning," *Journal of Information Security and Applications*, vol. 46, pp. 42–52, 2019, ISSN: 2214-2126. DOI: https://doi.org/10.1016/j.jisa.2019.02.008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214212618305866.

[14] J. Sakhnini, H. Karimipour, and A. Dehghantanha, "Smart grid cyber attacks detection using supervised learning and heuristic feature selection," in *Proceedings of the 2019 IEEE 7th International Conference on Smart Energy Grid Engineering (SEGE)*, Oshawa, ON, Canada, Aug. 2019, pp. 108–112.

[15] S. Ahmed, Y. Lee, S.-H. Hyun, and I. Koo, "Unsupervised machine learning-based detection of covert data integrity assault in smart grid networks utilizing isolation forest," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 10, pp. 2765–2777, 2019. DOI: 10.1109/TIFS.2019.2902822.

[16] M. Esmalifalak, L. Liu, N. Nguyen, R. Zheng, and Z. Han, "Detecting stealthy false data injection using machine learning in smart grid," *IEEE Systems Journal*, vol. 11, no. 3, pp. 1644–1652, 2017. DOI: 10.1109/JSYST.2014.2341597.

[17] C. M. Ahmed, J. Zhou, and A. P. Mathur, "Noise matters: Using sensor and process noise fingerprint to detect stealthy cyber attacks and authenticate sensors in cps," in *Proceedings of the 34th Annual Computer Security Applications Conference*, ser. ACSAC '18, San Juan, PR, USA: Association for Computing Machinery, 2018, pp. 566–581, ISBN: 9781450365697. DOI: 10.1145/3274694.3274748. [Online]. Available: https://doi.org/10.1145/3274694.3274748.

[18] Y. Shoukry, M. Chong, M. Wakaiki, *et al.*, "Smt-based observer design for cyber-physical systems under sensor attacks," *ACM Trans. Cyber-Phys. Syst.*, vol. 2, no. 1, Jan. 2018, ISSN: 2378-962X. DOI: 10.1145/3078621. [Online]. Available: https://doi.org/10.1145/3078621.

[19] D. Hadžiosmanović, R. Sommer, E. Zambon, and P. H. Hartel, "Through the eye of the plc: Semantic security monitoring for industrial processes," in *Proceedings of the 30th Annual Computer Security Applications Conference*, ser. ACSAC '14, New Orleans, Louisiana, USA: Association for Computing Machinery, 2014, pp. 126–135, ISBN: 9781450330053. DOI: 10.1145/2664243.2664277. [Online]. Available: https://doi.org/10.1145/2664243.2664277.

[20] K. N. Junejo and J. Goh, "Behaviour-based attack detection and classification in cyber physical systems using machine learning," in *Proceedings of the 2nd ACM International Workshop on Cyber-Physical System Security*, ser. CPSS '16, Xi'an, China: Association for Computing Machinery, 2016, pp. 34–43, ISBN: 9781450342889. DOI: 10.1145/2899015.2899016. [Online]. Available: https://doi.org/10.1145/2899015.2899016.

[21] S. Pan, T. Morris, and U. Adhikari, "Classification of disturbances and cyber-attacks in power systems using heterogeneous time-synchronized data," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 3, pp. 650–662, 2015. DOI: 10.1109/TII.2015.2420951.

[22] R. C. Borges Hink, J. M. Beaver, M. A. Buckner, T. Morris, U. Adhikari, and S. Pan, "Machine learning for power system disturbance and cyber-attack discrimination," in *2014 7th International Symposium on Resilient Control Systems (ISRCS)*, 2014, pp. 1–8. DOI: 10.1109/ISRCS.2014.6900095.

[23] M. Keshk, N. Moustafa, E. Sitnikova, and G. Creech, "Privacy preservation intrusion detection technique for scada systems," in *2017 Military Communications and Information Systems Conference (MilCIS)*, 2017, pp. 1–6. DOI: 10.1109/MilCIS.2017.8190422.

[24] S. Pan, T. Morris, and U. Adhikari, "Developing a hybrid intrusion detection system using data mining for power systems," *IEEE Transactions on Smart Grid*, vol. 6, no. 6, pp. 3104–3113, 2015. DOI: 10.1109/TSG.2015.2409775.