**Course name: Programming and Problem Solving for Data Analytics**
**Course code: 04-802 E**
**Instructor:** Dr. George Okeyo
**Final Project:** Machine Learning for Detecting Cyber-attacks and Problems in Industrial Control Systems
**Name:** Ariane SHIMIRWA
**AndrewID**: ashimirw
**Submission date:** 19th December 2022

# REPORT: BUILDING MACHINE LEARNING MODELS FOR DETECTING POWER SYSTEM DISTURBANCES

## 1. BACKGROUND

Power systems are known as networks of electrical devices used to supply, transmit, and utilize electric power. The primary goal of a power system is to be resilient by providing electricity to people nonstop.[1] When a power system is having disturbances caused by either natural events or man-made events challenging its resilience, it can be hard for the operators to take decisions and can cause serious repercussions to the economy, security, and households as well. This study aims at building machine-learning models for detecting power system disturbances and problems in Industrial Control Systems (ICS) using the dataset that is based on measurements related to normal, disturbance, control, and cyber-attack behaviors for a power system [2].

## 2. PROBLEM DESCRIPTION

Nowadays most power systems are connected to the internet which can expose the security and operation of their system and can result in man-made events (cyber-attacks). Cyber-attacks can be damaging as nature attacks. A paper written by Rajkumar et al. illustrated how cyber-attacks on power systems against digital substations protection systems may generate power and disconnect lines which can trigger cascading failures in the power grid and cause a partial or complete blackout.[3] Therefore power firms need to take effective countermeasures to prevent unauthorized access to their systems to prevent these threats and it is critical to accurately detect cyberattacks on ICS power system controllers when creating interventions to strengthen the systems' resilience.

## 3. METHODS

The data used was downloaded from Industrial Control System (ICS) Cyber Attack Datasets it contained 15 datasets and they were all merged. A random sample of 5% of all the merged datasets was selected to reduce the size and make it easy for computations and evaluation of the models.
The selected sample had an average of 2759 attack instances, 963 natural events, and 195 no-event instances used as classes for classification. The dataset was further explored and analyzed by detecting nan values, duplicates, outliers using visualizations, and a class imbalance was observed. After deep analysis and visualization, the data was preprocessed by handling outliers, transforming categorical data to numerical, scaling the data because some data points were far from each other, and the imbalanced classes were balanced using smote algorithm which uses oversampling to synthesize new minority instances between existing minority instances.[4]

After the data was ready for building machine learning models, the classification models were constructed using 10fold cross-validation method. The models used were logistic regression, k-Nearest neighbor, and random forest.

**Steps involved done in model building**

1. The models were built before making feature selection and evaluated

2. Feature selection was performed to reduce data dimension using the stepwise forward regression method which selects significant features in model construction based on their p-values.
3. The features selected by forward regression were then used to reconstruct the models (logistic regression, Random Forest, and KNN) and evaluated their performance before and after reducing the dimensionality of features.
4. The best model selected before and after feature selection was then used as a benchmark model to identify optimal hyperparameters that give the best estimator for it using Grid search and 10-fold cross-validation. The two models (benchmark model and tuned model) performances were then compared using visualization.

The performance metrics used for model evaluation were classification metrics Precision, Recall, and f1 score. These metrics were selected because our class distribution was uneven, and these metrics give a clearer view of how the classifier produces errors. Precision measures the positive predictive values, recall measures the true positive rate, and f1 score is the harmonic mean of precision and
recall. A value closer to 1 indicates a great performance.[5]

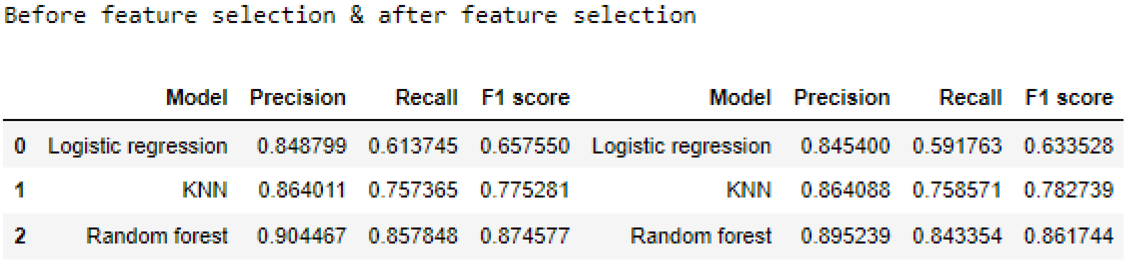
## 4. EXPERIMENTS AND RESULTS

### 1. Model creation before performing feature selection and after performing feature selection

The Models were constructed using 10fold cross-validation and using all the features, Random Forest was the best model among the other models with the highest precision, recall, and f1 scores tending to 1. After performing feature selection using forward regression, it selected 33 features as the significant features in the model performance. Thus, they were used to build our models.

Random Forest model was also the best model with greater values of the score tending to 1 than other models. As illustrated in Fig4.1 below shows the results of the two models before and after feature selection.

Fig4.1 Comparison of the performance of the models before using feature selection and after using it



```
Before feature selection & after feature selection
```

| | Model | Precision | Recall | F1 score | Model | Precision | Recall | F1 score |
|---|---|---|---|---|---|---|---|---|
| 0 | Logistic regression | 0.848799 | 0.613745 | 0.657550 | Logistic regression | 0.845400 | 0.591763 | 0.633528 |
| 1 | KNN | 0.864011 | 0.757365 | 0.775281 | KNN | 0.864088 | 0.758571 | 0.782739 |
| 2 | Random forest | 0.904467 | 0.857848 | 0.874577 | Random forest | 0.895239 | 0.843354 | 0.861744 |

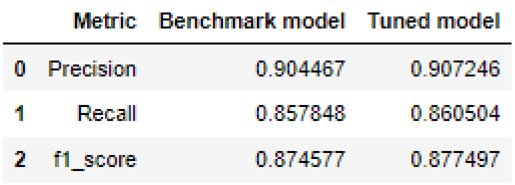
From Fig 4.1, comparing both models before and after feature selection, it appears that the random forest model before feature selection was the best model overall. Thus, making it **the benchmark model**. The random forest model before feature selection was the best model may be because feature selection selected only 33 features among 128 and lost contents of other important features that were important in model selection thus decreasing its performance.

### 2. Hyperparameter tuning the benchmark model

Tuning the hyperparameter benchmark in the set range, the optimal parameters for the benchmark model were the maximum death = 120 and the number of estimators (number of trees in a forest) = 30. The obtained parameters were then used in evaluating the performance of the tuned model together with the benchmark model.

<u>Result</u>

Fig 4.2.1Comparing the benchmark model and the tuned model



|   | Metric | Benchmark model | Tuned model |
|---|---|---|---|
| 0 | Precision | 0.904467 | 0.907246 |
| 1 | Recall | 0.857848 | 0.860504 |
| 2 | f1_score | 0.874577 | 0.877497 |

Fig4.2.1 shows that after tuning the benchmark model and building it using hyperparameters that give the best estimators the model performance slightly increases. It is more visual in fig 4.2.2 below.

Fig4.2.2 A graph showing comparison between the benchmark model and the tuned model

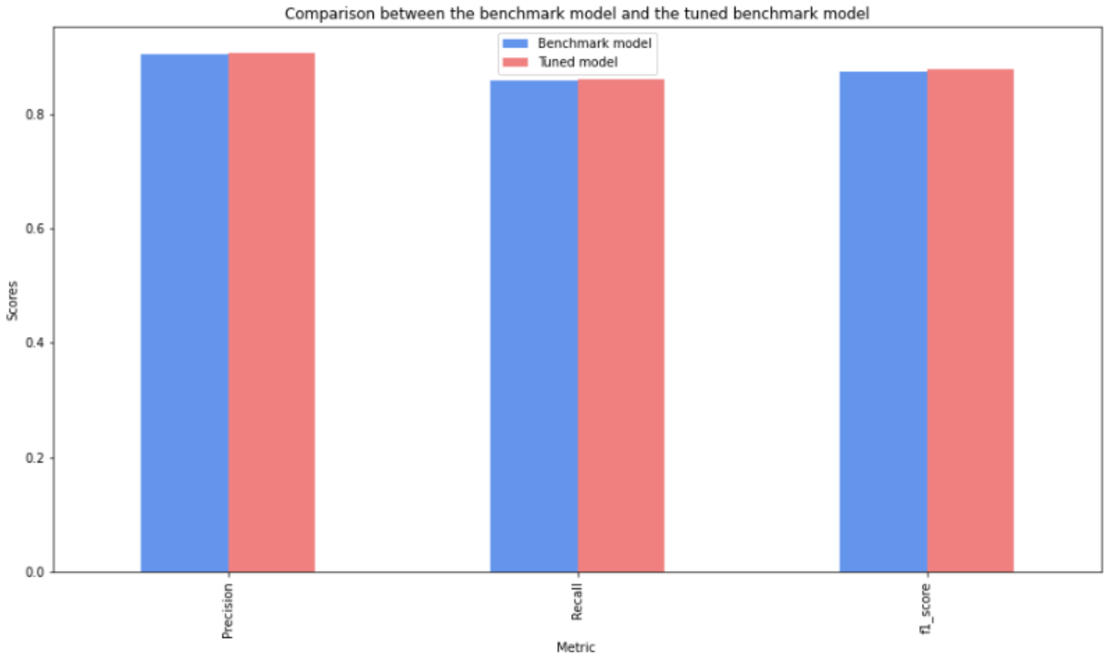


Fig 4.2.2. shows a slight change in the performance of the model after tuning it. This concludes that tuning the model to get the best estimators increases the performance of the model.

## 5. DISCUSSION AND CONCLUSION

The analysis made showed that hyperparameter tuning plays an important role in building the models because it gives the best performance compared to other methods. On the other hand, I observed that some datasets do not perform well after feature selection, and this may be because our data had many features, and feature selection selected a few of them dropping the other features. Those dropped feature contents that were lost decreased the performance of the model. Thus, some of the models used (Logistic and RF) were sensitive to feature selection on this particular dataset. Another observation is that classes were imbalanced which was going to mislead the classifiers to focus more on the large classes and ignore the small ones, thus balancing the classes made sure the accuracy provides a good general indicator of classifier performance.

To conclude, this analysis was made by applying machine learning algorithms to a power system based on measurements related to normal, disturbance, control, and cyber-attack behaviors using three classes attack events, and no events. The values of the recall that reflect true positives indicated that the classifier detected cyber-attacks most successfully and the f1 score that balances precision and recall on the positive class was high which indicated that the model performed well. Thus, applying machine learning methods on this dataset might be the best approach for ICS in detecting cyber-attacks on their systems.

## REFERENCES

[1] Contributors to Wikimedia projects. "Electric power system - Wikipedia." Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Electric_power_system (accessed Dec. 19, 2022).

[2] "Tommy Morris - Industrial Control System (ICS) Cyber Attack Datasets." Google Sites: Sign-in. https://sites.google.com/a/uah.edu/tommy-morris-uah/ics-data-sets?pli=1 (accessed Dec. 19, 2022).

[3] V. S. Rajkumar, M. Tealane, A. Ştefanov, A. Presekal and P. Palensky, "Cyber Attacks on Power System Automation and Protection and Impact Analysis," 2020 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe), 2020, pp. 247-254, doi: 10.1109/ISGT-Europe47291.2020.9248840.

[4]"ML | Handling Imbalanced Data with SMOTE and Near Miss Algorithm in Python - GeeksforGeeks." GeeksforGeeks. https://www.geeksforgeeks.org/ml-handling-imbalanced-data-with-smote-and-near-miss-algorithm-in-python/ (accessed Dec. 19, 2022).

[5] Beaver, J., Borges, R., Buckner, M., Morris, T., Adhikari, U., Pan, S., Machine Learning for Power System Disturbance and Cyber-attack Discrimination, Proceedings of the 7th International Symposium on Resilient Control Systems, August 19-21,2014, Denver, CO, USA.