Electronics and Computer Science
Faculty of Engineering and Physical Sciences
University of Southampton

Detection of Attacks on Power System Grids Report

Lukas Kakogiannos

Student ID: 32158998

May, 2024

Contents

1 Introduction

This report aims to analyse the machine learning models and techniques used to detect potential cyber-attacks on a power system grid's framework. A functional and secure framework is important for the following reasons:

- **Reliability:** The primary purpose of a framework is to ensure the reliable operation of the power grid. By monitoring the status of each component in real-time, operators can quickly identify and address any issues that arise, minimising downtime and ensuring a steady supply of electricity.
- **Cybersecurity:** With the increasing digitisation of power grids, cybersecurity has become a significant concern. A framework provides tools for detecting and mitigating cyber-attacks.
- **Safety:** A framework allows for quick detection and isolation of faults. This not only prevents damage to expensive equipment but also ensures the safety of personnel working on the grid.
- **Efficiency:** By allowing for real-time monitoring and control, the framework helps improve the efficiency of the power grid.
- **Maintenance:** The framework allows operators to perform maintenance activities safely. By manually tripping breakers, operators can isolate parts of the grid for maintenance without affecting the rest of the system.

For the purpose of this report, cyber-attacks can take two forms. The first is data injection, where an attacker changes values to parameters such as current, voltage, sequence components, etc., to imitate a valid fault. This attack aims to blind the operator and cause a blackout. The second form is remote tripping command injection, where a command is sent to a relay which causes a breaker to open. This can only be done once an attacker has penetrated the system's defences.

The goal is to develop two machine learning models that can effectively identify a cyber-attack from measurements provided by 4 PMUs (phasor measurement units). The two different tasks to be completed are outlined below:

- **Binary Classification Task [Zhou, 2021]:** Given a dataset with 6000 entries, the aim is to develop a machine learning model that can successfully distinguish between two scenarios, measurements indicating that the system is operating regularly with no issues, and measurements indicating a potential data injection attack.
- **Multiclass Classification Task [Géron, 2017]:** Given another dataset with 6000 entries, the aim is to develop a machine learning model that can successfully distinguish between three scenarios, measurements indicating that the system is operating regularly with no issues, measurements indicating a potential data injection attack, and measurements indicating a potential command injection attack.

2 Machine Learning Models

When deciding which algorithm to train on the dataset, three possibilities were considered, Gradient Boosting Algorithms (like XGBoost) [Natekin and Knoll, 2013], Support Vector Machine (SVM) [Hearst et al., 1998],

and Random Forest Classification (RFC) [Pal, 2005], with the two latter ones being implemented for further testing, the reasons for these choices are outlined below:

1. Effectiveness in High-Dimensional Spaces: SVM is known for its effectiveness in high-dimensional spaces and its ability to handle complex decision boundaries [Steinwart and Christmann, 2008]. It can handle both linear and non-linear classification through the use of different kernel functions such as linear, polynomial, or radial basis function (RBF).
2. Robustness: Random Forests are generally more robust to noisy data and less prone to overfitting compared to Gradient Boosting algorithms [Belgiu and Drăguţ, 2016]. This makes them a safer choice for many applications.
3. Computationally Efficient: SVM and RFC are often more computationally efficient than Gradient Boosting algorithms [Géron, 2017].

2.1 Pre-processing

An important step before feeding the data to the machine learning algorithms is to run it through a pipeline with pre-processing steps, for both tasks, the first step uses the StandardScaler estimator to standardise features by removing the mean and scaling to unit variance (displayed below) [Géron, 2017].

```
# Create a pipeline
pipe = Pipeline(steps=[('scale', StandardScaler()), ('rf', clf)])
```

Listing 1: Pipeline used for Random Forest Classifier

Depending on the model, standardising the features can lead to better performance, for instance, models that can be sensitive to the scale of the features, such as SVMs. Moreover, since StandardScaler is more sensitive to outliers, combining it with an RFC implementation can provide a robust way of dealing with them [Balabaeva and Kovalchuk, 2019].

2.2 Support Vector Machine

2.2.1 Binary Classification Task

At its core, SVM is a binary linear classifier, it works by finding the hyperplane that best separates the data into two classes [Hearst et al., 1998]. The "best" hyperplane is the one that maximises the margin between the closest points (called support vectors) of the two classes, known as the maximum margin hyperplane. One of the key advantages of SVM is its memory efficiency, however, it's worth noting that SVM can be sensitive to the choice of the kernel and its parameters [Steinwart and Christmann, 2008]. To mitigate this problem, a grid search algorithm can be implemented, it is a method used to optimise the performance of machine learning models by systematically searching through a specified parameter grid and determining the best combination of hyperparameters [Lerman, 1980]. Hyperparameters are parameters not learned during training but are set before training and significantly impact the model's performance and behaviour.

The hyperparameter grid is displayed below:

```
# Define the parameter grid
param_grid = {'svm__C': [0.1, 1, 10, 100],
              'svm__gamma': [1, 0.1, 0.01, 0.001],
              'svm__kernel': ['linear', 'rbf']}
```

Listing 2: Parameter grid used for grid search in SVM

The best parameters for the given dataset were the following:

{'svm__C': 100, 'svm__gamma': 0.1, 'svm__kernel': 'rbf'}

The results will be further discussed in section 3. Furthermore, SVM implementations for multiclass classification tasks require the use of additional techniques like one-vs-one or one-vs-rest [Steinwart and Christmann, 2008], this led to the implementation of alternative models, such as Random Forest.

2.3 Random Forest

The RFC is a powerful machine learning algorithm that creates many decision trees during the training phase, each tree is constructed using a random subset of the dataset and a random subset of features at each partition [Pal, 2005]. This randomness introduces variability among individual trees, reducing the risk of overfitting and improving overall prediction performance [Belgiu and Drăguţ, 2016]. In the prediction phase, the algorithm aggregates the results of all trees by voting, this process works for both binary and multiclass classification tasks [Géron, 2017].

2.3.1 Binary Classification Task

To implement the RFC, the standard RandomForestClassifier from sklearn.ensemble was used, similarly to the SVM implementation, a StandardScaler and a grid search algorithm are implemented to increase its effectiveness, the parameter grid used is displayed below:

```
# Define the parameter grid
param_grid = {'rf__n_estimators': [100, 200, 300],
              'rf__max_depth': [None, 10, 20, 30],
              'rf__min_samples_split': [2, 5, 10]}
```

Listing 3: Parameter grid used for grid search in RFC (binary classification)

The best parameters for the given dataset were the following:

{'rf__max_depth': 20, 'rf__min_samples_split': 2, 'rf__n_estimators': 300}

The results will be further discussed in Section 3.

2.3.2 Multiclass Classification Task

Due to the nature of the task of having three different possible scenarios, the effectiveness of the RFC implementation used for the binary classification task was bound to be reduced. However, with a simple change to the parameter grid, adding more possible hyperparameters and values, this problem was mitigated (shown below):

```
# Define the parameter grid
param_grid = {'rf__n_estimators': [100, 200, 300],
              'rf__max_depth': [None, 10, 20, 30],
              'rf__min_samples_split': [2, 5, 10],
              'rf__max_features': ['sqrt', 'log2'],
              'rf__min_samples_leaf': [1, 2, 4],
              'rf__bootstrap': [True, False],
              'rf__criterion': ['gini', 'entropy']
}
```

Listing 4: Parameter grid used for grid search in RFC (multiclass classification)

The best parameters for the given dataset were the following:

{'rf__bootstrap': False, 'rf__criterion': 'gini', 'rf__max_depth': 30,

'rf__max_features': 'log2', 'rf__min_samples_leaf': 1,

'rf__min_samples_split': 2, 'rf__n_estimators': 300}

The results will be further discussed in Section 3.

3 Evaluation

The first step for evaluation is to split the dataset into two smaller ones by using the train_test_split function, the dataset is split into a training set that is 80% the size of the original one, and a validation set that contains the remaining 20%. This is done so the model's performance can be tested on previously unseen data [Lever et al., 2016], a visual representation of the performances is displayed in the confusion matrices below:

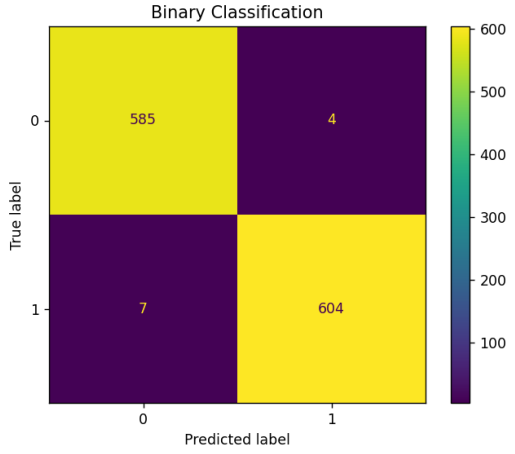

Figure 1: Confusion Matrix for Binary

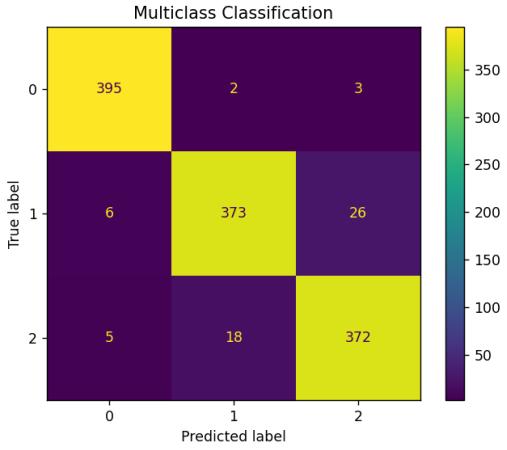

Figure 2: Confusion Matrix for Multiclass

From the matrices above, it can be observed that both classification tasks were successful, as the number of false positives (FP) and false negatives (FN) is near zero in almost all possible scenarios. Furthermore, using the number of true positives (TP), true negatives (TN), FP and FN, four evaluation metrics can be calculated [Géron, 2017, Islam et al., 2019]:

- Precision - The ability of the model to correctly identify positive instances among all instances it predicted as positive [Davis and Goadrich, 2006].
- Accuracy - The proportion of correctly classified instances out of the total [Wheeler and Calder, 2007].
- Recall - The ability of the model to capture all positive instances in the dataset.
- F1 Score - The harmonic mean of precision and recall, providing a balance between the two.

All four metrics range from 0 to 1, with 1 being the best possible value meaning the model predicts the values correctly 100% of the time [Zhou, 2021]. While the values for the SVM binary implementation were satisfactory, giving precision, accuracy, recall, and f1 scores of 0.97, the RFC implementation surpassed it by giving a near-perfect prediction (depicted in Figure 1), leading to precision, accuracy, recall, and f1 scores of 0.99.

Moreover, the RFC implementation for the Multiclass classification task is depicted in Figure 2, its precision, accuracy, recall, and f1 scores were 0.95, this was due to a few misclassifications between the two types of attacks.

4 Conclusion

In conclusion, the presented approaches adeptly tackle the issue of recognising potential cyber-attacks on a power system grid. The utilisation of the dataset facilitates a nuanced exploration of phasor measurement unit readings, enabling the formulation of two robust classification systems for distinguishing between regular activity, data injection attacks, and command injection attacks, it is safe to conclude from the confusion matrix for the Multiclass implementation that the readings for both types of attacks can be similar at times, leading to a small number of misclassifications. However, both models successfully recognise an attack over 99% of the time.

Moreover, the chosen machine learning algorithms, Support Vector Machines (SVM) and Random Forest Classification (RFC) prove suitable for the above classification tasks. Evaluation metrics, such as confusion matrices, accuracy, recall, precision and F1 scores, demonstrate the models' ability to discern between the three scenarios, with both algorithms exhibiting excellent performances, but RFC being the best one for both tasks.

Even though the binary classification is accurate over 99% of the time, there are still possible improvements to be made in future iterations, for instance, other algorithms such as neural networks could be explored, additionally, more sophisticated feature engineering could be performed to extract more informative features from the phasor measurement unit readings, in order to differentiate between the two types of attacks more effectively.

Finally, the code used can be found at the following link:

`https://colab.research.google.com/drive/1L_AI-7o4V5_CZEzNtxnv0YCTxkZQE2fF?usp=sharing`

References

[Balabaeva and Kovalchuk, 2019] Balabaeva, K. and Kovalchuk, S. (2019). Comparison of temporal and non-temporal features effect on machine learning models quality and interpretability for chronic heart failure patients. *Procedia Computer Science*, 156:87–96. 8th International Young Scientists Conference on Computational Science, YSC2019, 24-28 June 2019, Heraklion, Greece.

[Belgiu and Drăguţ, 2016] Belgiu, M. and Drăguţ, L. (2016). Random forest in remote sensing: A review of applications and future directions. *ISPRS journal of photogrammetry and remote sensing*, 114:24–31.

[Davis and Goadrich, 2006] Davis, J. and Goadrich, M. (2006). The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240.

[Géron, 2017] Géron, A. (2017). *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media.

[Hearst et al., 1998] Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J., and Scholkopf, B. (1998). Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4):18–28.

[Islam et al., 2019] Islam, M. Z., Liu, J., Li, J., Liu, L., and Kang, W. (2019). A semantics aware random forest for text classification. page 1061–1070. Association for Computing Machinery.

[Lerman, 1980] Lerman, P. (1980). Fitting segmented regression models by grid search. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 29(1):77–84.

[Lever et al., 2016] Lever, J., Krzywinski, M., and Altman, N. (2016). Points of significance: Model selection and overfitting. *Nature Methods*, 13:703–704.

[Natekin and Knoll, 2013] Natekin, A. and Knoll, A. (2013). Gradient boosting machines, a tutorial. *Frontiers in neurorobotics*, 7:21.

[Pal, 2005] Pal, M. (2005). Random forest classifier for remote sensing classification. *International journal of remote sensing*, 26(1):217–222.

[Steinwart and Christmann, 2008] Steinwart, I. and Christmann, A. (2008). *Support vector machines*. Springer Science & Business Media.

[Wheeler and Calder, 2007] Wheeler, D. C. and Calder, C. A. (2007). An assessment of coefficient accuracy in linear regression models with spatially varying coefficients. *Journal of Geographical Systems*, 9:145–166.

[Zhou, 2021] Zhou, Z. (2021). *Machine Learning*. Springer.