Use of the SVM Model for Cyber Attack Detection in a SCADA Dataset

J. Lapierre, Z. Soomro, M. Greenfield, F. Koch

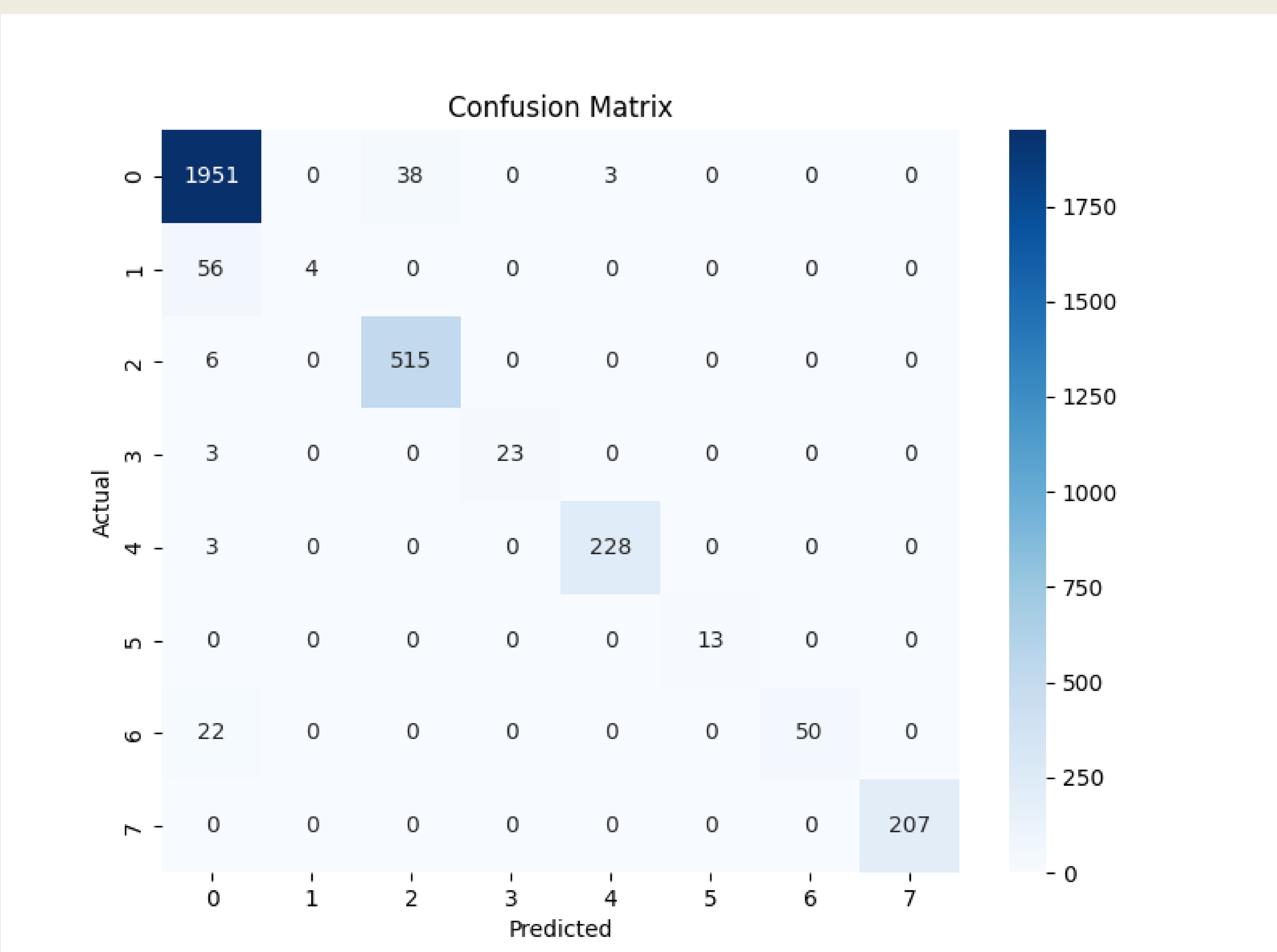
University of Wisconsin - Milwaukee

OBJECTIVES

- Utilize the support vector machine (SVM) model of artificial intelligence to efficiently analyze data within a SCADA system
- Use that model to distinguish between regular and irregular data points, as well as between benign and malicious activities in the data stream to predict potential attacks on the system.
- Evaluate the ability of SVM in correctly assessing those anomalies compared to other methods.

APPROACH

- Use an existing dataset from Tommy Morris' Industrial Control System Cyber Attack Datasets(4) as the training set for the SVM model. Namely, the gas pipeline data set was chosen for initial training / analysis.
- Clean the dataset by removing outliers, scaling features,
- Train the SVM model using the gas pipeline data set to identify attack types.
- Test the accuracy of the model by feeding other datasets from Tommy Morris' page and comparing the algorithm's results to the results column in the dataset.
- Evaluate the SVM algorithm's ability to detect attacks and speculate on ways to improve its performance.



CONTACT INFORMATION

Logan Lapierre - lapierr3@uwm.edu
Zachary Soomro - zsoomro@uwm.edu
Magan Greenfield - magan@uwm.edu
Felix Koch - frkoch@uwm.edu

- **Normal (0):** Standard operations.
- **Naïve Malicious Response Injection (NMRI) (1).**
- **Complex Malicious Response Injection (CMRI) (2).**
- **Malicious State Command Injection (MSCI) (3).**
- **Malicious Parameter Command Injection (MPCI) (4).**
- **Malicious Function Code Injection (MFCI) (5).**
- **Denial of Service (DOS) (6).**
- **Reconnaissance (Recon) (7).**

METHODOLOGY

Data preprocessing:

Used a python script (clean_data.py) to perform the following operations:

- Remove outliers in the 1st or 99th percentile range.
- Convert the results column of the data set into integers for easier training and result analysis.
- Scaled the numeric features of the dataset to have a mean of zero and a standard deviation of one.

Training the SVM model:

Fed the processed data into a python script (train_svm.py). The script randomly splits the processed data into a training subset (70% of the dataset) and a testing subset (the remaining 30%). After training the model, the predicted results of the the testing subset was compared to its actual values and compiled in a classification report. The model is then saved to a .pkl file for further evaluation.

Testing Accuracy:

Fed the generated model more processed datasets from Morris' database, except without splitting them into a training set. The predictions were compared to the results just like they were for the initial training set.

Evaluation:

The train_svm.py script evaluates precision, recall, F1-score, and support for each attack type in the original data set.

- Precision: The percentage of correctly predicted results per attack type. Gauges the model's ability to identify attacks.
- Recall: The percentage of correctly predicted positive results out of all expected positive results. Essentially gauges the model's aversity to false negatives.
- F1-score: An average between precision and recall, which balances the two values into a more comprehensive metric.
- Support: The number of entries per attack type present in the testing set.

In addition to these four metrics, train_svm.py also generates the average and weighted average for each metric, and an overall accuracy score. Using the results compiled from the initial dataset as well as others tested from Morris' database, we can derive a comprehensive result for the SVM model.

RESULTS

- Initial testing of the SVM model using the initial training / testing set provided an overall accuracy score of 0.96, over 3122 entries tested.
- The model indicated a slight bias towards falsely identifying ordinary data as an attack, rather than falsely identifying an attack as ordinary data.
- More testing will be required on other datasets for more comprehensive results.

	Precision	Recall	F1-score	Support
0	0.96	0.98	0.97	1992
1	1.00	0.07	0.12	60
2	0.93	0.99	0.96	521
3	1.00	0.88	0.94	26
4	0.99	0.99	0.99	231
5	1.00	1.00	1.00	13
6	1.00	0.82	0.82	72
7	1.00	1.00	1.00	207

CONCLUSION

•With an accuracy score of 0.96, the SVM model proves very effective at identifying attacks in a SCADA dataset.

•Other models may be similarly or more effective, with an Adaboost and Jripper combination boasting an average precision score of over 0.99(2).

•Performance could be improved for the SVM model in several ways:

- Feeding the model additional datasets.
- Preprocessing the data more aggressively.
- Combining the SVM model with a second algorithm that specifically handles the ordinary data points marked as attacks by the SVM model to improve the system's overall recall score.

BIBLIOGRAPHY

(1) Maglaras, L. A., & Jiang, J. (2014). Intrusion detection in SCADA systems using machine learning techniques. *2014 Science and Information Conference (SAI)*, 626-631. IEEE.

(2) Borges Hink, R. C., Beaver, J. M., Buckner, M. A., Morris, T., Adhikari, U., & Pan, S. (2014). Machine learning for power system disturbance and cyber-attack discrimination. *2014 IEEE International Conference on Big Data (Big Data)*, 757-762.

(3) Hadžiosmanović, D., Bolzoni, D., & Hartel, P. (n.d.). Towards securing SCADA systems against process-related threats. *University of Twente*.

(4). Morris, T. (n.d.). Industrial Control System (ICS) Cyber Attack Datasets.