

University of Alabama in Huntsville

**LOUIS**

---

Honors Capstone Projects and Theses

Honors College

---

5-9-2024

## Performance Evaluation and Comparison of Machine Learning Models in Anomaly Detection of a SCADA ICS

Hugh Charles Vessels  
*University of Alabama in Huntsville*

Follow this and additional works at: <https://louis.uah.edu/honors-capstones>

---

### Recommended Citation

Vessels, Hugh Charles, "Performance Evaluation and Comparison of Machine Learning Models in Anomaly Detection of a SCADA ICS" (2024). *Honors Capstone Projects and Theses*. 915.  
<https://louis.uah.edu/honors-capstones/915>

This Thesis is brought to you for free and open access by the Honors College at LOUIS. It has been accepted for inclusion in Honors Capstone Projects and Theses by an authorized administrator of LOUIS.

**Performance Evaluation  
and Comparison of  
Machine Learning  
Models in Anomaly  
Detection of a SCADA ICS**

by

**Hugh Charles Vessels**

**An Honors Capstone  
submitted in partial fulfillment of the  
requirements for the Honors Diploma to**

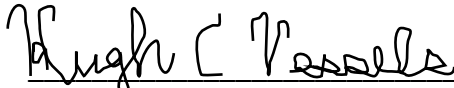
**The Honors College**

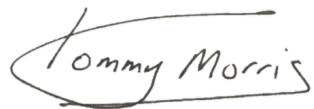
of

**The University of Alabama in Huntsville**

**05/09/2025**

**Capstone Project Director: Dr. Tommy Morris**

 05/09/2024  
Student (signature) Date

 05/10/2024  
Director (signature) Date

\_\_\_\_\_  
Department Chair (signature) Date

\_\_\_\_\_  
Honors College Dean (signature) Date



Honors College Frank Franz Hall  
+1 (256) 824-6450 (voice) +1 (256) 824-7339 (fax)

### Honors Thesis Copyright Permission

**This form must be signed by the student and submitted with the final manuscript.**

In presenting this thesis in partial fulfillment of the requirements for Honors Diploma or Certificate from The University of Alabama in Huntsville, I agree that the Library of this University shall make it freely available for inspection. I further agree that permission for extensive copying for scholarly purposes may be granted by my advisor or, in his/her absence, by the Chair of the Department, Director of the Program, or the Dean of the Honors College. It is also understood that due recognition shall be given to me and to The University of Alabama in Huntsville in any scholarly use which may be made of any material in this thesis.

Hugh C Vessels  
Student Name (printed)

Hugh C Vessels  
Student Signature

05/09/2024  
Date

## Table of Contents

Dedication	2
Abstract	3
Introduction	4
Chapter 1: Machine Learning Models	7
Chapter 2: Anomaly Detection	10
Chapter 3: Anomaly Detection Models	11
Chapter 4: Current Advantages and Disadvantages of Popular Anomaly Detection Machine Learning Models	14
Chapter 5: Future Work	17
Chapter 6: Reference Lists	19
Conclusion	20

## **Dedication**

I would like to dedicate my honors capstone to my faith, family, and friends. I would also like to dedicate my honors capstone to my project director, Dr. Tommy Morris, and all the professors and other members of the UAH community who have supported me. Finally, I would like to acknowledge and thank my fellow senior design group members: Liam Coleman, Paul Choe, Hunter Bolling, Luke Wathen, and Neil Ollenburger.

## **Abstract**

I plan for my honors capstone to be an extension of my senior design capstone. My senior design capstone involves creating cyber attacks to test on the Digital Twin of a SCADA system and record the response of the Digital Twin in a data logger. Eventually, collecting enough response information to form a data lake. What I plan to add with my honors capstone is to implement 2 to 3 different machine learning models (possibly a mix of supervised vs unsupervised models) to look for anomalies in the data lake and measure the performance of these models in anomaly detection and compare their effectiveness. Ideally, instances of when cyber attacks occurred on the Digital Twin should be treated as anomalies. The models I am currently considering are isolation forest, support vector machines (SVM), and autoencoders. I will implement the machine learning models I choose by using add-on Python libraries to run the training and test data through.

I can check the effectiveness of these models by cross referencing which attacks succeeded (should be referenced as an entry in the data logger) with which models detected said attacks/anomalies.

My findings should help narrow down the options for machine learning models that could be used in anomaly detection of SCADA response information by demonstrating which models were more effective. By narrowing down machine learning models, I can provide a basis for the CCRE to start from when they implement their own machine learning-based anomaly detection system for the Digital Twin SCADA system (reason for project).

## Introduction

One trend that is not going away is the rise in cybersecurity threats. Most often when people think about cyber threats, they think of cyber attacks targeted at large industries like banking, technology companies, etc. However, there is one area in industry that is also seeing its fair share of attacks on its systems and that is supervisory control and data acquisition (SCADA) systems. SCADA systems are useful in monitoring and managing everyday industrial control systems that form the backbone of society. SCADA is used in industrial control systems like water plants, nuclear reactors, power grids, etc. (critical everyday infrastructure).

Gone are the days of believing that any computer-based systems are impervious to vulnerabilities, as evident in the number of increasing zero-day vulnerabilities (therefore, increasing the adoption of Zero Trust architecture in response). SCADA systems are “vulnerable to many attacks including attempted break-in, penetration by legitimate user, leakage by legitimate user, Trojan horse, virus, logic bomb, denial-of-service attack, and so on” (Yang et al., n.d.). In my senior design project, our own cyber attacks were able to install malicious chrome extensions that randomized displayed values, shutdown entire programmable logic controllers (PLCs), removed all connections from the human machine interface (HMI) to the PLCs through firewall rule manipulation, and performed a man-in-the-middle attack between slave PLCs and a master PLC (ultimately changing values represented for respective slave PLC on the HMI). All this to say, my senior design group was successful in exploiting vulnerabilities in the SCADA system we were provided for our project (our first time creating cyber attacks). In addition, one vulnerability that is always a concern is the insider threat, as “SCADA

systems are moving over to standard protocols, and the deregulation of many industries (especially the electricity industry) makes their control systems more vulnerable to manipulation by malicious insiders” (Bigham et al., 2003).

Due to the prevalent vulnerabilities that can potentially be exploited in these systems, there is a need for better detection of anomalies that could potentially be malicious. Due to the relatively recent and successful rise in machine learning applications for anomaly detection, I decided to make my capstone about which machine learning models can be most effective in anomaly detection from a performance stand point. I define performance for machine learning models in anomaly detection to be how accurate the models are at identifying any deviations from normal data patterns while minimizing the amount of both false positives and false negatives.

The data I intended to use for said machine learning models is from my senior design project. The data collected would be from the SCADA system’s response to our cyber attacks as well as data from its normal operations. Due to delays in my senior design project that include: unforeseen complexity in automating our cyber attacks, breaking virtual machines, reinstalling the SCADA system multiple times, creating our own data loggers for our SCADA system, implementing cleanup programs that essentially remove attack artifacts and allow the attack to be redeployed; data collection from my senior design project happened very late into the semester that this honors capstone was due. Therefore, I was not able to get the data in time to implement the models as previously discussed in the abstract for my honors capstone. Due to delays in data collection and not having enough time to implement the machine learning models for anomaly detection, my honors capstone will shift focus to a more theoretical



approach on how I think certain machine learning models would perform if I had been able to collect data in time to implement them.

## **Chapter 1: Machine Learning Models**

To understand how anomaly detection works, we must first understand how machine learning models work and what they are. Machine learning models are algorithms that are trained with input data and then tested with another group of data (test data) to recognize patterns and/or make predictions. There are many types of machine learning models: supervised learning, unsupervised learning, semi-supervised learning, reinforcement learning, etc. In general, machine learning models can be either classified as supervised or unsupervised. The distinction between the two is that supervised models use labeled input data to make their predictions or recognize patterns. In contrast, unsupervised models do not require their input to have labels in their data.

Originally, the data that would have been collected from my senior design project would have been an example of supervised data as it has labels on its columns of data. However, in my research, I found unsupervised machine learning models with some success in anomaly detection. Therefore, this paper will consider both unsupervised and supervised models in its performance evaluation (advantages and disadvantages) in anomaly detection. In addition, this paper is intended to review models for the general use case. Therefore, not all data generated will be supervised in terms of applications outside of my own senior design project. So, it is better overall to consider the major types of machine learning models when doing the performance evaluation.

Figure 1 shows an example of the data I would have been able to collect if my senior design project had no delays. As you can see, the data is labeled. Naturally indicating this data would be good as input for supervised learning models. This data includes the power, current, and voltage values from six different programmable logic controllers (PLC) on a SCADA system (both the user interface data and the data used in a MATLAB Simulink model) used in my senior design project.

Ground Truth Utility Voltage	HMI Time Slot	HMI Current Mode	HMI Generator Power	HMI Generator Current	HMI Generator Voltage
220	5/2/2024 9:29	9	3821	17	217
219	5/2/2024 9:29	9	3175	14	219

**Figure 1: Potential Input Data from Senior Design Project**

Figure 2 below shows a CSV file of cyber attack data also from my senior design project. In the figure below you see the name of the attacks (ex: Extension), whether they were successful (0) or not (1), and what time they were executed. The idea behind these two data sets (represented in Figure 1 and Figure 2) was to feed the data from the SCADA system (user interface and Simulink data, shown in Figure 1) into machine learning models. The machine learning models would then determine the anomalies among the Figure 1 data set and compare the anomalies with the Figure 2 attack data to see if the anomaly was an actual attack executed or just an anomaly in the SCADA system. The comparison would be between the Simulink data and the user interface to see if there has been any manipulation of the values given by the Simulink model (would show as different values in the user interface). The following comparison would be to try to see if an attack was executed at the same time (compare timestamps) that the Figure 1 dataset

shows differences in its user interface data and its Simulink data. If there is a difference between the Simulink and user interface data as well as an attack executed at that time, then we could conclude that the anomaly was an attack. From there, I would analyze which machine learning models had better overall performance in identifying anomalies that were also cyber attacks created in my senior design project. It is important to note that the data is incomplete, like the data in Figure 1, due to reasons mentioned previously.

3/29/2024 14:26	Extension	1
4/4/2024 18:08	IptableAttack	0
4/4/2024 18:08	AITMAttack	1
4/4/2024 18:10	IptableAttack	0
4/4/2024 18:10	PLC_shutdown	0

**Figure 2: Attack Data to Cross Reference Anomalies from Machine Learning Models**

## Chapter 2: Anomaly Detection

Anomaly detection as described in machine learning models is the identification of data points that fall outside of the normal range. The normal range is different based off each machine learning model, as each machine learning model learns differently with the training data on what is considered as normal. It is important to note that not every data point flagged as an anomaly should be considered as a cyber attack against the system. In many cases, the system itself non-maliciously produces the anomaly instead of an external actor, which is why it is important to train the model with as much data as possible before the model starts overfitting.

Machine learning models that are overfitting mean that they start making too many generalizations about the data, which leads to less than satisfactory results and predictions. Ways to prevent overfitting include: stopping the training data phase before there is too much statistical noise coming from the data set, regularization to remove the non-statistically significant factors, data augmentation to slightly change the input data so it can be considered unique, etc. Ultimately it is believed in intrusion detection that cyber attack behavior will cause effects that vary significantly from normal operations in a system or device. Deviation from normal behavior (a.k.a anomaly) in machine learning models and data sets is usually a calculated statistical number of standard deviations from a range of values the model determines as normal or baseline. In the next chapter, we will go over what models I have found that are commonly used for anomaly detection.

### **Chapter 3: Anomaly Detection Models (Omar et al., 2013)**

The two most important components in a machine learning based project are the model and the data used in the model. Therefore, it is important to note that “the general architecture of all anomaly based network intrusion detection systems (A-NIDS) methods is similar” (Omar et al., 2013). These stages essentially can be classified as data collection, model training, and detection. In data collection, you are collecting representative data from the target system. The representative data is then processed as either training or test data. In model training, the model is given training data as input to form its analysis of data patterns and create predictions from said data. In detection, the models are then given test data to assess these predictions to determine which anomalies were predicted or tested. Next, is analyzing the results of the model and its ability to detect anomalies. Ideally, the machine learning model’s performance in anomaly detection would be measured by an F-score, which is a metric used to determine predictive performance of a model (however, this requires the models to be implemented). In addition, for machine learning model performance in anomaly detection, you would use a receiver operating characteristic (ROC) curve to determine the number of false positives versus true positives.

In my research, there are generally five common models used for anomaly detection: one-class support vector machines (SVM), isolation forest, autoencoders, k-nearest neighbor, and long short-term memory (LSTM). One-class SVM, isolation forest, and autoencoders are unsupervised machine learning models. K-nearest neighbor and LSTM models are supervised machine learning models. One important thing to note for all machine learning models (regardless of type) is the machine learning models will always

have some form of statistical error. Therefore, each model will strive to identify the most anomalies caused by attacks while trying to balance false negatives and false positives from anomalies in the system itself.

One-class support vector machine is a specific version of an SVM. One-class SVM's initial premise is that the input data is normal (not anomaly or outlier). Classifying data as normal for this model leads to creating a normalcy region to later determine data points outside of that region (outlier boundary, anomaly). The SVM model then uses a process called margin maximization that tries to create a buffer of safety around the normalcy region and the outlier data. This buffer of safety essentially acts as the cutoff for the statistical deviation from normal to anomaly. This buffer is determined when feeding input data into the model by using a hyperparameter (specifies details of the model's learning process). This parameter is a value that essentially acts as the acceptable number of standard deviations from the initial normal data points before the data point becomes an outlier. Therefore, adjusting this parameter directly adjusts the number of potential false positives or false negatives. In other words, one can adjust this parameter like a dial in order to get the best fit possible for the respective model and data.

Isolation forest models operate under the initial premise that all of the data are anomalies. Isolation forest, like the name, isolates individual data points. First, the isolation forest creates many isolation trees. A random split value is chosen and then the model proceeds to isolate the individual points. The model isolates points by determining if the data point is less than or greater than the random split value. If the data point is greater, it becomes a right node off the random split value. If the data point is lower, it becomes a left node off the random split value. The isolation forest then takes all the data

points and finds their node path length. It is generally accepted that shorter node paths are more likely considered to be anomalies.

Autoencoders are a specific model among a group of models called neural networks. There are three different layers in how autoencoders work in anomaly detection: encoding, bottleneck, and decoding. The idea behind autoencoders is to encode the input data, force it through a medium (bottleneck where data is being compressed), decode it, and see what values are different from before they were encoded. The errors in being able to reconstruct a data point after decoding is what is classified as an anomaly. Autoencoders train on data whose premise is to be mostly normal.

K-nearest neighbor model is similar to other density-based techniques/models where each data point is viewed in relation to a range of its nearest neighbors (k closest) data points. In other words, each data point is evaluated as an anomaly or not based on the classification and values of the closest data points to it. If 9 out of 10 data points say you are not similar to them, then you are considered an anomaly (this is an example). The acceptable amount of standard deviation in the model is determined by the user.

The LSTM model is a form of a neural network. The LSTM model for anomaly detection operates by looking at large amounts of previous values and making a prediction from that on what the next value should be. The difference in how long the model took to get the value it predicted is compared to the standard deviation. If the model's prediction is within said standard deviation, then the data point is considered normal. If the model's prediction is outside said standard deviation, then the data point is considered an anomaly.



## **Chapter 4: Current Advantages and Disadvantages of Popular Anomaly Detection Machine Learning Models (Elmrabit et al., 2020)**

Since data collection was cut short, I will not be able to input data from my senior design project into my own machine learning models as previously described. Instead, I will theorize what potential results I would have seen given the general performance characteristics of each type of model.

The advantages of one class SVM include efficiency with multi-factored data sets, high memory efficiency, works well with standard deviations to make clear which data points are anomalies. The disadvantages of one class SVM is that it does not scale well with large amounts of data per data set, can underperform if not given enough data points, and does not work well with noise in the data set.

The advantages of isolation forest are that it is easy to use, scales well with large data sets, and has fast computational abilities compared to other detection algorithms. The disadvantages of isolation forest include overfitting (too many isolation trees) and sensitivity to the number of trees in the models as well as other factors.

The advantages of autoencoders include reducing noise from the input data set and as a means of data augmentation (reducing overfitting due to consistent nonduplicating results). The disadvantages of autoencoders include the decoding process losing information from the input data set, which could affect the accuracy of the results (false positives and false negatives).

The advantages of k-nearest neighbor are little or no time to train the model as well as ease of use. The disadvantages of k-nearest neighbor include not being able to scale with large data sets and work with multi-factored data sets.

The advantages of LSTM include that it can track long term patterns and bidirectional data processing. The disadvantages of one class LSTM include being very resource dependent and being prone to overfitting.

Based on my research into strengths and weaknesses of these models, I believe the isolation forest model would be the highest performing with data used from my senior design project. I believe this because this model works well with larger data sets, is fast, easy to implement, and is the only model discussed here that is designed specifically for anomaly detection. I think one-class SVM model would be the second highest performing model because it does a great job of identifying anomalies but usually in smaller data sets. Therefore, one would have to preprocess the data into more manageable chunks, which takes time and computational power. I think LSTM would be the third highest performing model because it does a great job of looking at large data sets and identifying patterns given the whole scope of previous data, making it very accurate. However, it is not easy to implement and takes too many resources. I think the autoencoder model would be the fourth highest performing model because it reduces the chances of overfitting but still loses a good amount of information, which becomes more of a problem with large data sets. I think the k-nearest neighbor model would be the fifth highest performing model because its advantage and disadvantage are in its simplicity. It's the easiest to use among the models, however, it is also not meant for the size and type of data being fed in as input. Therefore, my recommendation for any future work in implementing a machine-learning based anomaly detection mechanism for the SCADA system used in my project would be an isolation forest model. In (Elmrabit et al., 2020), they found in their analysis that the "Random Forest (RF) algorithm achieves the best performance in terms of accuracy,

precision, Recall, F1Score and Receiver Operating Characteristic (ROC) curves on all these datasets". The data sets used to test their machine learning models include data sets with attacks on industrial control systems (ICS, SCADA is a form of ICS). Despite the study not using an isolation forest as one of its models for anomaly detection, one can see the success that decision-tree based models have in anomaly detection. The differences between random forest models and isolation forest models include: random forest models are supervised, built for classification and regression, and are less scalable than isolation forests.

## Chapter 5: Future work

Often the best practice in machine learning model related projects, such as anomaly detection, is to create/implement their own models. In other words, the best results tend to come from machine learning models that are customized to your project, data set, variables/factors, etc. If I had additional time, I would like to be able to create/customize my own machine learning model to see if I can get better performance out of a custom model than one already available as an easy to load in Python library.

In addition, if I had more time to work or expand on this capstone, I would have looked into implementing my own version of an anomaly detection machine-learning based intrusion detection system (IDS). An anomaly detection machine-learning based intrusion detection system would be very beneficial, I believe, to the CCRE and their use of multiple types of digitized SCADA systems. As using this IDS on different types of systems and data would not only improve the tools performance, but it would also provide the CCRE a proof of concept for when they go to implement their own IDS with their own custom machine learning models. In addition, I believe if I could use a “defense-in-depth” approach to the intrusion detection system I could layer the IDS with multiple types of anomaly detection models that could help pick up anomalies that the other models would not find on their own.

Since SCADA systems are becoming more connected to the internet, the concept of security through obscurity “completely falls apart in SCADA networks, as special-purpose hardware is replaced by COTS servers and proprietary communication protocols by the TCP/IP stack” (Barbosa, 2014). This means that it is easier for attackers to find out information on SCADA systems as well as test their attacks on similar available products.

Therefore, making it even more important that attacks can be detected in these systems in order to prevent attackers from causing catastrophic failure. However, it is also important to realize that intrusion detection is a passive security tool. Meaning that the attacks will continue to occur even with this measure in place. Therefore, it is paramount that anomaly-based intrusion detection systems be paired with preventive cyber tools, like intrusion prevention systems (IPS). Another potential expansion to this capstone would be to integrate more active and preventative cyber tools and software along with the anomaly detection machine-learning based intrusion detection system. This addition would allow the CCRE to both detect and defend against cyber attacks on their digital SCADA installations.

## Chapter 6: Reference List

- Barbosa, Rafael Ramos Regis. "Anomaly detection in SCADA systems: a network based approach." (2014).
- Bigham, John, David Gamez, and Ning Lu. "Safeguarding SCADA systems with anomaly detection." In *Computer Network Security: Second International Workshop on Mathematical Methods, Models, and Architectures for Computer Network Security, MMM-ACNS 2003, St. Petersburg, Russia, September 21-23, 2003. Proceedings 2*, pp. 171-182. Springer Berlin Heidelberg, 2003.
- Elmrabit, Nebrase, Feixiang Zhou, Fengyin Li, and Huiyu Zhou. "Evaluation of machine learning algorithms for anomaly detection." In *2020 international conference on cyber security and protection of digital services (cyber security)*, pp. 1-8. IEEE, 2020.
- Omar, Salima, Asri Ngadi, and Hamid H. Jebur. "Machine learning techniques for anomaly detection: an overview." *International Journal of Computer Applications* 79, no. 2 (2013).
- Yang, Dayu, Alexander Usynin, and J. Wesley Hines. "Anomaly-based intrusion detection for SCADA systems." In *5th intl. topical meeting on nuclear plant instrumentation, control and human machine interface technologies (npic&hmit 05)*, pp. 12-16. 2006.

## **Conclusion**

This capstone reviewed the importance of cyber threats affecting SCADA systems, a potential solution through anomaly detection with machine learning models, and which models I believe would perform the best if used with the previously mentioned data. From my evaluation of five different machine learning models, I believe the isolation forest model would perform the best out of all those reviewed for anomaly detection of the SCADA system used from my senior design project. There are thousands of different machine learning models available now, which I hope my research has aided in narrowing down some reliable options in implementing the right anomaly detection models in securing tools, systems, etc. The future of securing SCADA, as well as other systems, is in machine learning-based anomaly detection. They can be used in intrusion detection systems as well as other cyber tools and anti-malware to better identify if an attack or any other potentially harmful activity is occurring.