



Detecting Anomalies in Industrial Control Systems

Danyelle Loffredo, University of New Mexico

Project Mentor: Ryan Adams, Org 5628

Problem Statement:

- Detecting intrusions in industrial control systems (ICS) that are masked as legitimate commands can be difficult to detect, but important to stop. Very few labelled datasets on such attacks are available to developers of analytics that could detect these attacks.

Objectives and Approach:

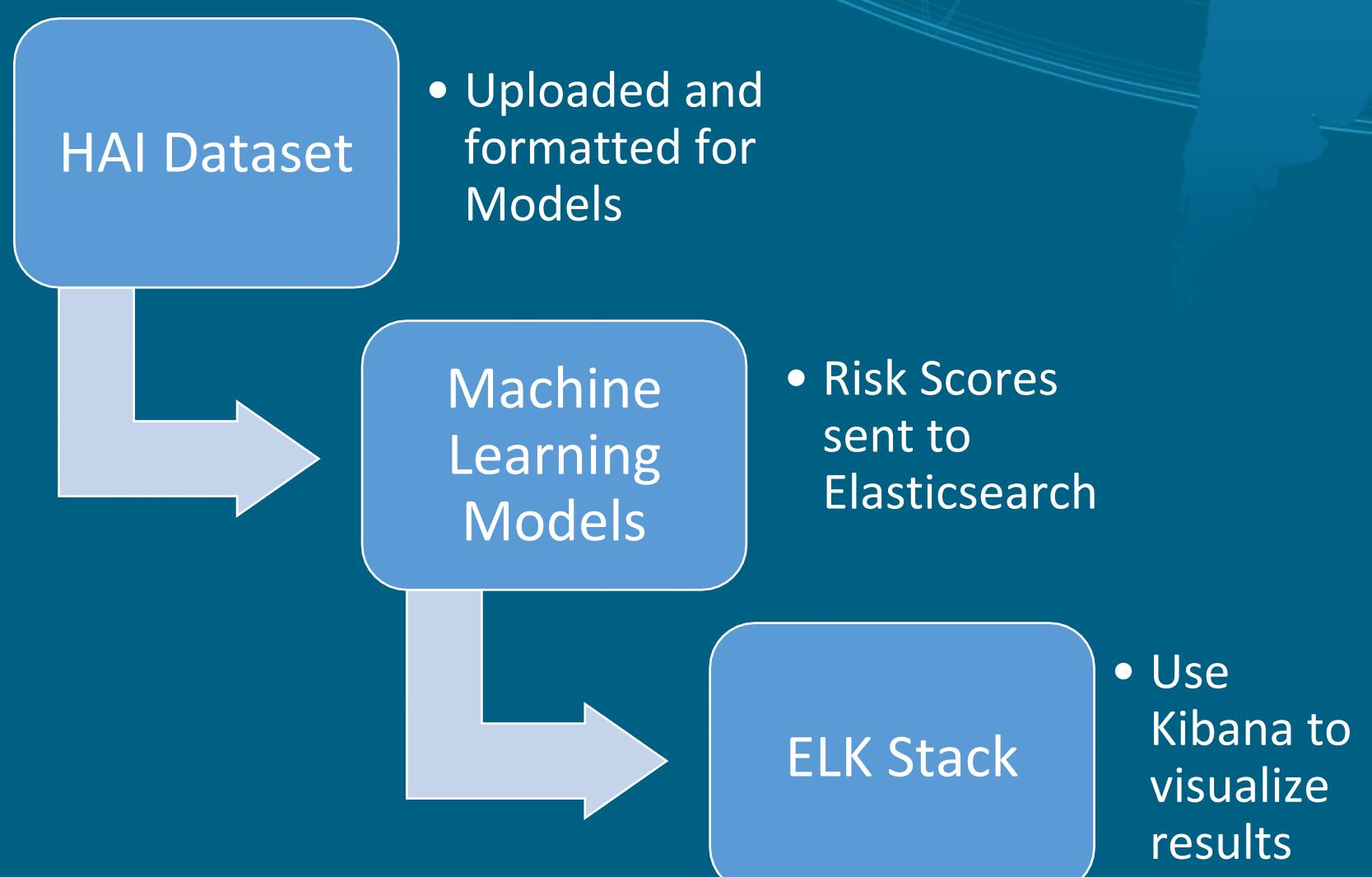
- Use machine learning techniques such as artis and principal component analysis (PCA) to analyze ICS data points for anomalies and process understanding
- Test the models on one of the few open source data sets, specifically the HAI dataset
- Present data in visualizations that organizations can use to take action against active events or to understand incidents in a forensics view

HAI Dataset Overview from Documentation

Version	Data Points	Normal Dataset			Attack Dataset		
		Files	Interval	Size	Files	Attack Count	Interval
HAI 21.03	78 points/sec	train1.csv	60 hours	100 MB	test1.csv	5 attacks	12 hours
		train2.csv	63 hours	116 MB	test2.csv	20 attacks	33 hours
		train3.csv	229 hours	246 MB	test3.csv	8 attacks	30 hours
					test4.csv	5 attacks	11 hours
					test5.csv	12 attacks	26 hours
HAI 20.07 (HAI1.0)	59 points/sec	train1.csv	86 hours	127 MB	test1.csv	28 attacks	81 hours
		train1.csv	91 hours	98 MB	test1.csv	10 attacks	42 hours

<https://github.com/icsdataset/hai>

Process Breakdown



Results:

- Dataset dimensions were appropriately reduced using PCA
- Testing the models on the HAI dataset showed strong correlations between upticks in risk scores and confirmed attacks
- Results can be exported to Elasticsearch and viewed as visualizations in Kibana

Example dashboard from Kibana using HAI dataset



Impact and Benefits:

- Successful implementation of these techniques can lead to advancing the state and understanding of ICS analytics
- Advancements in ICS analytics using the HAI dataset could encourage organizations to create more open source datasets