Feature Classification for Control System Devices

M Rayhan Ahmed Mithu, Mike Rogers, Denis Ulybyshev, Rajesh Manicavasagam, Rima Asmar Awad
Department of Computer Science
Tennessee Technological University, Cookeville, USA
{mmithu42, mrogers, dulybyshev, rmanicava42}@tntech.edu, awadrl@ornl.gov

Abstract

Control systems are used to automate industrial processes, smart grids, and smart cities. Unfortunately, cyber attacks on control systems are on the rise. Additionally, control systems lack the plethora of tools available for commodity systems for forensic investigation. An important step towards the proper forensic investigation is to analyze device memory. To assist in identifying features of device memory, we present a machine learning-based technique that integrates ontology information for feature classification in a control system device's memory.

Introduction

A control system is a set of interconnected devices that coordinate to control dynamic systems in industry, the power grid, and smart cities. Control systems can be found anywhere that automation is needed to increase productivity, consistency, and safety. Control devices monitor and control other devices and tend to require little energy, but are computationally slower than commodity hardware. Historically, due to their lack of processing power, these devices lack the security measures of commodity hardware because they were typically isolated and lacked connection to a network. However, controls systems are now often deployed in uncontrolled environments, such as buildings accessible by many employees, or out in the field. Furthermore, these devices have become more remotely accessible through wide area networks and the Internet. The increased accessibility has also increased the opportunity for attacks, which can have severe, even life threatening, consequences.

Forensic analysis is used to determine characteristics of a cyber attack. However, forensic analysis needs to be done as quickly and thoroughly as possible to mitigate cost and damage. Machine Learning (ML) techniques are becoming more widely used in computer forensics because they can decrease the time of the investigation (Qadir and Varol 2020). Unfortunately, few tools exist for identifying features in a device memory which is an important step of forensic analysis.

In this paper, we present a methodology for automatic identification of features in a device memory using raw device data, such as device memory dumps or log files, and an ontology that describes the control process.

To describe our methodology, our goals are as follows.

- Define example sets of ontologies for established control systems. In this paper, we incorporate two such examples.
- Show how these ontologies could be used with our feature classification methodology to classify the artifacts from raw unlabelled data that can be obtained from logs and/or device memory dumps.
- Show how these ontologies with the selected features can be used to classify device artifacts for device states.
- Evaluate our feature classification methodology and make a determination about which algorithms work for our example domains and datasets.

Related Work

Closely related to our work are the areas of *ontology in control systems*, *feature extraction and classification*, and *forensics*. Ontologies are formal representations and definitions of the running processes of a domain that also provide knowledge about the relations between artifacts of the domain (Gruber and others 1993). Some of the works using ontology in control system are: Küçük et al. (Küçük and Arslan 2014) domain ontology for wind power applications by learning the concepts and properties of this domain from Wikipedia articles.

Melik-Merkumians et al. proposed an ontology-based fault diagnosis for control systems in (Melik-Merkumians, Zoitl, and Moser 2010), they used an ontology to represent the core concepts and instances of a minimalistic tank model.

Work in feature extraction and classification includes Christ et al. who proposed an approach named FRESH - Feature Extraction based on Scalable Hypothesis tests, which is a highly parallel feature filtering system (Christ, Kempa-Liehr, and Feindt 2016). Tsang et al. (Tsang and Kwong 2005) implemented an Independent Component Analysis (ICA) with an Ant Colony Clustering Model (ACCM) to extract features and reduce dimensionality for intrusion detection. Both of these works aim to enhance feature extraction with component analysis and dimensionality reduction to improve the clustering algorithms, and their approaches attempt to identify the important features in the dataset. In our case, we incorporate additional information extracted from the ontology to improve the performance of our clustering algorithms.

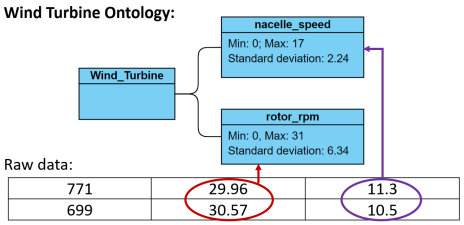

Figure 1: Relation between sensor data and ontology.

Forensic analysis in control systems can be challenging because shutting down the control system to collect forensic data is not practical. A necessary initial phase is to examine and identify the control system (Spyridopoulos, Tryfonas, and May 2013). Two phases of forensic methodology described by Stirland et al. (Stirland et al. 2014) are identification and preparation. Our work can be used by the investigators to complement these steps and use the collected data to quickly get familiar with the system under investigation.

Feature Classification Workflow

We use a two phase workflow for feature classification from the raw data. The first phase labels the data according to the information in the ontology, and the second phase applies ML feature classification algorithms.

Ontology and Labeling

In the first phase of our workflow, we collect raw, unlabeled data. Typically, this raw data will come from the device log file or debugging interfaces. However, for these initial experiments, we used freely available datasets. The authors of these datasets have already labeled them. For our purposes, we removed the labels from the data so that they represent the raw data collected from devices. The next step is to incorporate the ontology information. This preprocessing associates characteristics and behaviors in the ontology to the data values in the raw data. It results in a labeled data file whose labels are identifiers for the characteristics and behaviors from the ontology file. The labeling algorithm, as shown in Algorithm 1, takes as input the raw sensor data and the ontology. It then parses the ontology into a map of labels and predicates with tolerances. Then, each item in the sensor data is matched against the ontology by evaluating the predicate, and if the predicate evaluates to TRUE for all the sensor values for that sensor, then the data value is labeled with the corresponding label.

For example, in the ontology file, as shown in Figure 1, one of the value range is 0 to 17 which represents the sensor value of the nacelle speed. If the data values for a particular memory location all adhere to this behavior, then all the data values in the raw data are annotated with a label ”nacelle speed” that corresponds to this behavior according to an expected tolerance or deviation.

Feature Classification

In the second phase of our workflow, the labeled data file is used in the classification process, as shown in Figure 2.

Algorithm 1: Phase 1 - incorporating the ontology

```
function add_ontology (X, Y);
  Input  : raw_sensor_data (X), ontology (Y)
  Output : ontology_incorporated_data (Z)
for i = 0 to Y_length do
    for j = 0 to X_length do
        if Y[i].predicate applied to X[j] is TRUE then
            append (Y[i].label + X[j]) to Z;
        end
    end
end
return Z;
```

We have used both supervised and unsupervised learning in our experiments. For the unsupervised learning *K-means* clustering algorithm was selected for this experiment. The clustering algorithm takes the ontology incorporated data and assigns cluster labels to the data. The assigned clusters in this experiment represent different types of sensors. The next step is to use this labeled data for supervised learning. The supervised classification algorithms use the labeled data to create a prediction model. The data is split into training and test datasets. Once the training phase finishes, the prediction phase attempts to predict sensor data values in raw, unlabeled input data.

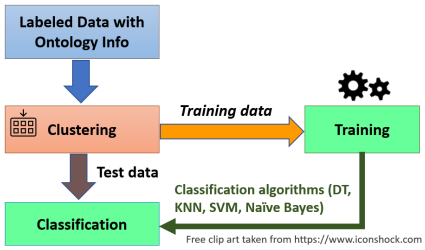

Figure 2: Phase 2 - clustering and classification.

In our experiments, to determine the effect of incorporating ontology information into the datasets, we compared the following classification algorithms: *Decision Trees*, *K-Nearest-Neighbors (KNN)*, *Support Vector Machines (SVM)*, and *Näive Bayes*. We chose these algorithms because they are in common use and have the most readily accessible libraries. We will incorporate more algorithms in our study in the future.

Experimental Results

We ran our experiments on the two datasets using various clustering and predictions algorithms, both with and without incorporating an ontology, and compared the results.

Experimental Setup

We applied our workflow in two different experiments on the WADI dataset (SUTD 2015) and the wind farm dataset(Passos et al. 2017). For the WADI dataset, we used 7 different types of sensors from the raw sensor data that had 192,000

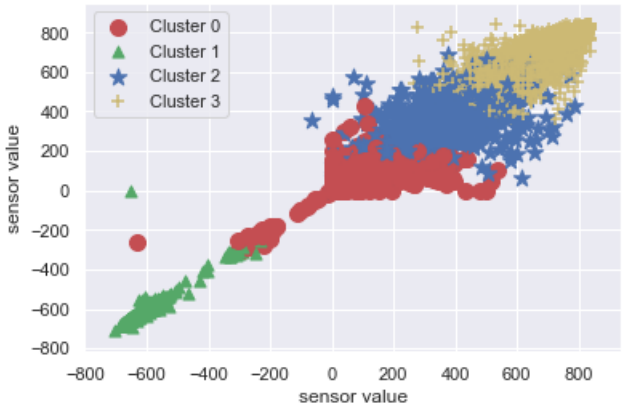

Figure 3: K-Means clustering w/o ontology data (Wind Farm).

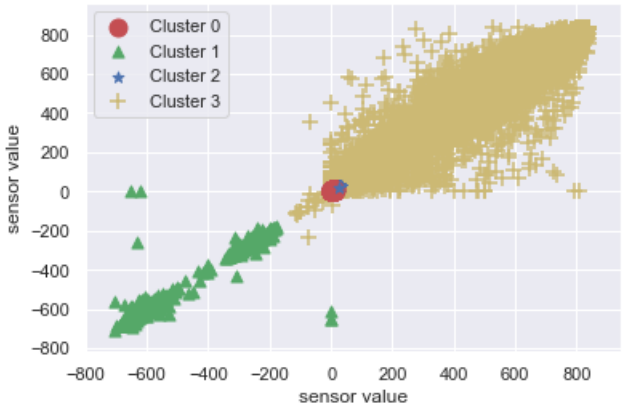

Figure 4: K-Means clustering with ontology (Wind Farm).

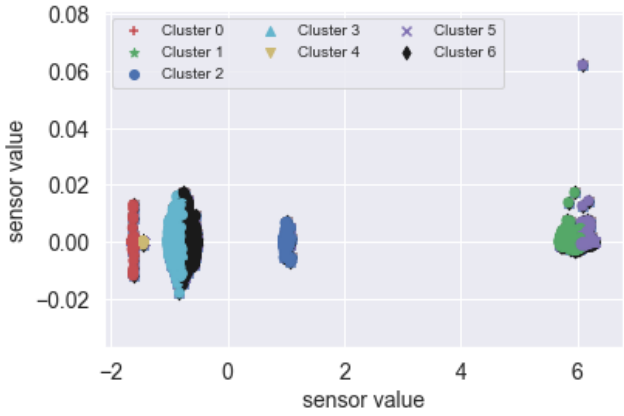

Figure 5: K-Means clustering w/o ontology (WADI).

records. The dataset consisted of an average of 14% to 15% sensor data for each of the 7 sensors. Each sensor in this dataset had 4 attributes. For the wind farm dataset, 4 types of sensor values were used to create a dataset of raw sensor data with 105,100 records. Each sensor had approximately 25% of sensor data. There are five attributes for the sensors.

As discussed in section "Feature Classification Workflow", the workflow labeled the data with ontology information in a preprocessing step, clustered the data, and then trained the classifier. After the workflow completed, we tested the classifier by running it and computing metrics for its statistical performance. Then in each experiment we ran the training and testing phases again, but excluded the preprocessing step in the workflow so that the ontology was not used. Running the workflow without the ontology information allows to compare the classifier on both raw and labelled data.

Evaluation

All of the classification algorithms performed similarly, according to their F-measure. However, the F-measure for some of the raw data was below 0.6. We found that the performance of the classification highly depends on the performance of the clustering algorithm.

Clustering without the ontology information results in poor clustering performance and, thus, poor classification performance. For example, Figure 3 shows four clusters for the wind farm dataset labeled as 0-3. Each cluster represents a different sensor. This figure illustrates running the workflow on the unlabeled data. Some of the raw data values are associated with the incorrect sensors, which behave similarly, but not exactly the same. The similar behavior makes distinguishing between the sensors difficult for the clustering algorithm. For example, in Figure 3 cluster 0 represents the values of the wind speed sensor which have a maximum value of 17.1 and minimum of 0. Similarly, cluster 0 represents the values of the active power sensor and the values for this sensor range from 0 to 840. As a result, the similarity of the values for the active power sensors contributed to clustering some of them as the wind speed sensor values. However, as shown in Figure 4, which incorporates the ontology, the clustering improved significantly. Though, some values of the active power sensors fall in the same range as the values of the wind speed sensors, the ontology supplies the needed information to tell them apart. The clustering algorithm uses this information to differentiate two sensors successfully.

Including ontology information improved clustering, and thus improved the classification performance metrics across all the classification algorithms. Precision increased from as little as 61% to 99% for the wind farm dataset and from 62% to 79% for the WADI dataset. The F-measure was higher than .75 for all cases with .99 being highest. Figure 5 clustering results of K-Means algorithm with and without ontology data for the WADI dataset. and Figure 6 show the Principal component analysis was used to reduce the dimensionality of the dataset. To evaluate the performance of the K-Means clustering algorithm we also implemented DBSCAN for the clustering phase. Figure 7 and Figure 8 illustrate the DBSCAN clustering performance on the wind farm dataset with and without ontology. Due to the nature of the dataset being highly sparse between different ranges, DBSCAN labels a lot of the data as noise (cluster 6). As a result, even though the performance of the algorithm improves when using the

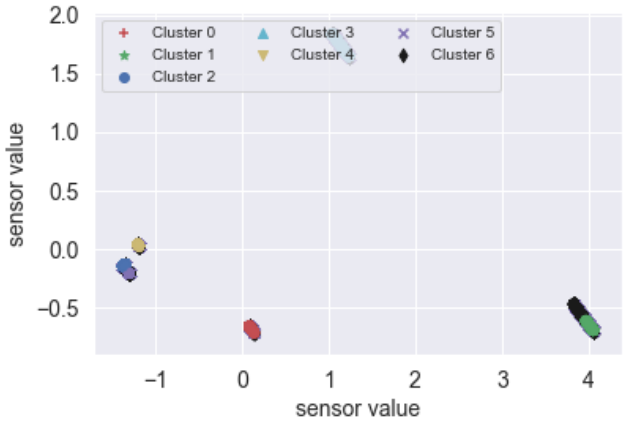

Figure 6: K-Means clustering with ontology (WADI).

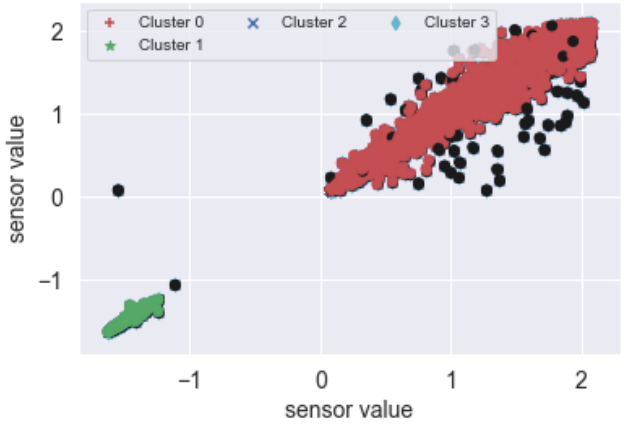

Figure 7: DBSCAN clustering w/o ontology (Wind Farm).

dataset with the ontology, the K-Means algorithm produces the best clustering results for the classification phase.

Conclusion

In this paper, based on the need for tools and methods for forensic analysis in control systems, we have presented a methodology that uses domain knowledge, clustering, and feature classification techniques to label features of raw memory dumps. Incorporating the domain knowledge in the form of ontologies significantly improves the performance of clustering and thus classification of the sensors in the control system devices. Our results are encouraging because having reliable device state data can help to discover and investigate cyber attacks at early stages in a control system.

Acknowledgments

We thank Zishan Ahmed Onik and Caleb Huck for their help with the project implementation. We thank Anthony Palmer and Bradley Northern for their feedback. We also thank the Center for Energy Systems Research at Tennessee Technological University for equipment and funding support.

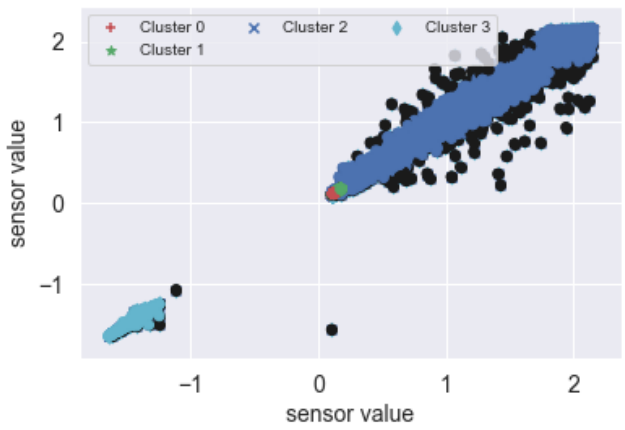

Figure 8: DBSCAN clustering with ontology (Wind Farm).

References

Christ, M.; Kempa-Liehr, A. W.; and Feindt, M. 2016. Distributed and parallel time series feature extraction for industrial big data applications. *arXiv preprint arXiv:1610.07717*.

Gruber, T. R., et al. 1993. A translation approach to portable ontology specifications. *Knowledge acquisition* 5(2):199–220.

Küçük, D., and Arslan, Y. 2014. Semi-automatic construction of a domain ontology for wind energy using wikipedia articles. *Renewable Energy* 62:484–489.

Melik-Merkumians, M.; Zoitl, A.; and Moser, T. 2010. Ontology-based fault diagnosis for industrial control applications. In *2010 IEEE 15th Conference on Emerging Technologies & Factory Automation (ETFA 2010)*, 1–4. IEEE.

Passos, J.; Sakagami, Y.; Santos, P.; Haas, R.; and Taves, F. 2017. Costal operating wind farms: two datasets with concurrent scada, lidar and turbulent fluxes. `https://zenodo.org/record/1475197#.Xx9RvZ5KhPY`. Accessed: 2020-11-16.

Qadir, A. M., and Varol, A. 2020. The role of machine learning in digital forensics. In *2020 8th International Symposium on Digital Forensics and Security (ISDFS)*, 1–5.

Spyridopoulos, T.; Tryfonas, T.; and May, J. 2013. Incident analysis digital forensics in scada and industrial control systems. In *8th IET International System Safety Conference incorporating the Cyber Security Conference 2013*, 1–6.

Stirland, J.; Jones, K.; Janicke, H.; Wu, T.; et al. 2014. Developing cyber forensics for scada industrial control systems. In *InfoSec 2014. The Society of Digital Information and Wireless Communication*, 98–111.

SUTD. 2015. Dataset characteristics of water distribution (wadi). `https://itrust.sutd.edu.sg/itrust-labs_datasets/dataset_info/`. Accessed: 2020-11-16.

Tsang, C.-H., and Kwong, S. 2005. Multi-agent intrusion detection system in industrial network using ant colony clustering approach and unsupervised feature extraction. In *2005 IEEE intl. conf. on industrial technology*, 51–56. IEEE.