EAI Endorsed Transactions
on Industrial Networks and Intelligent Systems

Research Article ICST.ORG

A Novel Intrusion Detection Mechanism for SCADA systems which Automatically Adapts to Network Topology Changes

Barnaby Stewart[1], Luis Rosa[2], Leandros A. Maglaras[1,*], Tiago J. Cruz[2], Mohamed Amine Ferrag[3], Paulo Simões[2], Helge Janicke[1]

[1]School of Computer Science and Informatics De Montfort University Leicester, UK
[2]University of Coimbra, Coimbra, Portugal
[3]Department of Computer Science, Guelma University, Algeria

Abstract

Industrial Control Systems (ICS) are getting more vulnerable as they become increasingly interconnected with other systems. Industrial Internet of Things(IIoT) will bring new opportunities to business and society, along with new threats and security risks. One major change that ICS will face will be that of the dynamic network topology. Changes in the network architecture will affect the performance of the ICS along with the efficiency of the security mechanisms that are deployed. The current article investigates how changes in the network architecture of a supervisory control and data acquisition (SCADA) system affect the performance of an Intrusion Detection System IDS that is based on the One class Support Vector Machine (OCSVM). Also the article proposes an adaptive mechanism that can cope with such changes and can work in real time situations. The performance of the proposed adaptive IDS is tested using traces from a Hybrid ICS testbed with a dynamic topology.

Received on 28 November 2016; accepted on 14 January 2017; published on 01 February 2017

Keywords: Intrusion Detection Systems, Support Vector Machines, Adaptive Mechanisms

doi:10.4108/eai.1-2-2017.152155

1. Introduction

Industrial Control systems (ICS) have both a greater need for security and a more difficult environment in which that security can be implemented, when Compared to Information Technology (IT) systems. This is becoming ever more critical as ICS become increasingly connected to other systems (both ICs and IT) and, inevitably, the internet. The term SCADA is traditionally associated with the subset of ICS known as Wide Area Control systems but more recently is being used as synonymously with ICS as a whole. For the purposes of this article we will be intending the traditional use of the term. Industrial Internet of Things (IIoT) is still in an early stage bringing new opportunities for the business and the society along with new threats and security risks. As Sadeghi et al. stated in [1], due to high connectivity and complexity of these new systems, thorough approaches that deal with safety and security risks are needed. The necessity of cyber physical security is rising and traditional methods may not be effective any more.

The need for security in SCADA systems is much higher than for the majority of computer systems due to the potential impact and consequences of service degradation or failure. Despite this, at the time when most older systems were developed, their isolation and extensive use of proprietary technologies was often considered sufficient safeguard against interference [2, 3]. Additionally, security on the whole was a far lower priority for all computer systems, and the overriding priority for SCADA was reliability.

Support Vector Machines (SVM) provide a viable method for very quickly analysing and classifying data in order to provide an intrusion detection function. The speed of SVMs is critical in SCADA systems due to their distributed nature and the need for high speed detection and response as well as the ability to have minimum impact on the performance of the system itself. The traditional support vector machine algorithm, which uses positive and negative samples to train an SVM model, is suitable for solving such classification problems and

*Corresponding author. Email: leandros.maglaras@dmu.ac.uk

providing fast indication of any anomaly that happens in the system. However, special characteristics of SCADA and other industrial control systems which are fewer abnormal samples, high dimensions of data, and strong correlation make the development of suitable IDS more demanding. Moreover most of data in industrial control system belong to normal communication behavior, and fault or critical state data are rare to find. Based on these observations, during recent years a lot of Intrusion Detection Systems that are based on the One Class Support Vector Machine (OCSVM) core are proposed [4–7]. OCSVM has only one category of data and simply tries to determine if new data belongs to that category or not. This allows OCSVMs to potentially detect new or unknown anomalies as it is not attempting to categorise any data as bad according to known attack prole, but simply tries to classify it as not good. These IDSs can be used in isolation or can be combined with other security mechanisms in order to provide a higher level of security [8].

Nowadays, as ICS systems are becoming increasingly connected (both to other systems and to the internet) as recognised in Yang et al [3], their security is becoming ever more important. In particular, Kim [2] notes that not only are SCADA systems more connected to the internet, but are also increasingly being implemented using shared Internet Protocol (IP) infrastructure and even the internet itself for establishing communication links. While most research focuses on the increased risks associated with these developments, it is important to note the importance of these changes to business in order to "reduce costs and increase efficiency" [9]. Many issues facing the implementation of security in SCADA have been identified by contemporary research:

- The need for reliability frequently overrides security considerations. This can make it very difficult to implement standard good practice, such as frequent patching.
- Lack of encryption in older communication protocols (plain text frequently used).
- Loss of obscurity caused by the adoption of widespread, well-documented protocols as well as the use of off the shelf SCADA systems [10]. Although obscurity is not a security mechanism in itself, its loss may facilitate attacks.
- The need for continuous operation makes it very difficult to update, modify, and maintain components of the system on the fly.
- Significantly longer lifespan for systems, potentially taking both hardware and software beyond their supported lifespan.

Moreover, SCADA security has specific characteristics and constrains which require a domain-specific approach. In-line security mechanisms (such as certain network IDS deployments) or host-level security tools (such as anti-virus) are unadvised because of the potential latency impact or the introduction of single points of failure in the critical communications path. Moreover, the increased sophistication of attacks against ICS infrastructures means that cyber-security cannot solely rely on supervised, pattern-based detection algorithms to ensure ongoing security monitoring. This situation requires complementary approaches for dealing with rogue threats, providing an adequate balance between its maintenance effort and detection robustness. The paper is organized as follows. Section 2 provides an overview of the most relevant literature concerning Intrusion Detection Systems. In Section 3, an overview of OCSVM is presented. Section 4 discusses the effects of network architecture changes on an IDS. In section 5.2, an overview of the used HEDV is given. Section 6 presents an initial testing of an OCSVM based IDS on different datasets.Section 7 describes a detailed design of the adaptive IDS. Section 8 presents a system performance evaluation. Finally, the conclusion is delivered in Section 10.

2. Intrusion Detection Systems

Many techniques for implementing IDS in a SCADA environment have been proposed. The two primary approaches are model based and machine learning. The model based approach is the more traditional of the two. and tends to result in fewer false positives but is also more likely to miss unknown attacks [4, 5]. The machine learning approach is more prone to generate false positives but is superior at detecting novel attack vectors.

2.1. Model based

Model based systems use detailed knowledge of the protocols and behaviours used within a system as well as details of known attacks to formulate rules which can identify both unexpected behaviour and known bad behaviour. This type of system is unlikely to recognise unknown attacks and requires frequent updates to the signatures to remain viable. The system proposed by Yang et al [3] comprises a number of model based methods including Access Control Whitelists; Protocol Based Whitelists and Behaviour Based Rules). The proposed system appears to have a greater capability for recognising unknown attacks than most model based system, but the study does not go into detail as to the time costs of properly setting up and maintaining such a system.

2.2. Neural Networks

Neural networks are a method of processing which mimics the manner in which biological systems, such

as the brain, operate. This is generally implemented in the form of neurons (or nodes) which exchange data. Multiple input nodes can receive data, process and pass it on to further nodes until an output node is reached. By calculating the cost of paths and modifying the behaviour of individual nodes it is possible for the network to adapt and learn (back propagation). By modelling complex relationships between the inputs and outputs of the system, it is possible to perform sophisticated pattern recognition. The main drawback of neural networks, as noted by both Pandit and Dudy [11] and Wang et al [12] is that they can take a long time to train and are difficult to scale.

2.3. Genetic Algorithms

This is a method for determining the optimal solution to a problem from a pool of potential solutions. In an evolutionary manner, poorly performing solutions are eradicated, leaving the best performing solutions standing. Essentially survival of the fittest. This is not an ideal technique for providing an IDS solution on its own, but is an excellent method for complementing and refining other machine learning systems to improve their accuracy and performance. Kim et al [13] proposes a system for supplementing an SVM based IDS with a Genetic Algorithms (GA) to ensure that the system maintained the most optimal detection model. The GA is used to detect both the optimal feature set as well as the optimal kernel and parameters. This can improve both the accuracy and the speed of the IDS. Recently a lot of similar approaches have been proposed with promising outcomes [14, 15].

2.4. Hierarchical Clustering

This method involves generating a dendrogram, which is a tree like structure representing clusters of data as differentiated by a chosen metric. This is a rapid means of categorisation and can be used to augment other systems. An example of this is given in Maglaras and Jiang [16] where k-means clustering is used recursively to categorise the outliers detected by the SVM process in order to reduce the number of false positives (in the form of severe alerts). Other similar works that combine clustering with a Bayes classifier [17] or nearest neighbors [18] also exist in the literature.

3. Support Vector Machines - OCSVM

SVMs provide a method for rapidly categorising data. The initial stage is the creation of a model using training data which can then be used to categorise new data. A defining characteristic of SVMs is the use of a kernel function to map data into a higher dimensional feature space such that the categories can be separated by hyperplanes. The SVM process iteratively determines the optimal hyperplane for distinguishing categories. The optimal hyperplane will have the largest margin between itself and the nearest data points and those data points which coincide with the margin are the support vectors.

It is possible to have multiple categories into which the data can be allocated, however when labeling of data is not possible One Class SVMS (OCSVM) can be used, which has a single category and simply determine if new data belongs to that category or not. This allows OCSVMs to potentially detect new or unknown anomalies as it is not attempting to categorise any data as 'bad' according to known attack profile, but simply to identify it as 'not good' [7, 19].

As there are no data points from a second class, the standard method for one class SVM is to treat the origin as the perfect class 1 vector and determine the optimal distance from the origin at which a data point no longer belongs to that class. Finding appropriate values for the calculation of this model is crucial in achieving the optimal result. This process is frequently improved by the use of other machine learning techniques to determine superior values for this algorithm.

3.1. Disadvantages

The main drawbacks of one class SVMs are false positives and over fitting, and as recognised by Maglaras & Jiang [16] and Wang et al [12]. False positives can result from rare but legitimate traffic which may not have been represented in the training data. Over fitting is a problem with the generation of the model where the boundary for categorisation is too tightly constrained to the test data. This can cause valid outliers to register as out of class (i.e. bad).

3.2. Examples of use

The CockpitCI Framework detailed in Cruz et al [20] uses a number of separate OCSVMs which are individually modelled for different parts of the ICS. The output of these is aggregated by a Main Correlator before being reported to the Security Management Platform. It is of note that the framework uses the Intrusion Detection Message Exchange Format (IDMEF) for interchange of data (RFC 4765). This is a message format to standardise the exchange of Intrusion Detection related information based on plain Extensible Markup Language (XML). Security features such as integrity or confidentiality are achieved by exchanging those messages via a dedicated Event Bus.

3.3. Suitability of OCSVMs in IDS

These systems all demonstrate that OCSVM is not a sufficiently refined tool to implement an effective IDS on its own but is highly valuable when coupled with other methods, especially as an integrated part of a larger

framework. It is likely that while anomaly detection provides opportunity to detect unknown attacks it will be necessary to combine them with signature and ruled based IDS components to achieve the greatest accuracy. Yang et al [3] have shown that multiple techniques can be used to create a highly effective IDS and Maglaras et al [4] argue that model based systems alone are insufficient. Wang et al [12] crucially note that a greater understanding of the range of SCADA applications and protocols is required to achieve truly effective model based IDS components.

4. Effects of Network Architecture Changes

We have established that, relative to IT systems, IC systems (most notably SCADA), have both a greater need for security and a more difficult environment in which to provide it. Support Vector Machines have been shown to be an effective tool for providing intrusion detection, although they lack effectiveness in isolation and are far more powerful when used in conjunction with other methods. Maglaras et al [4] argue that rule based IDS are ineffective, however the system proposed by Yang et al [3] shows that they can be implemented very effectively, although the maintenance demands are likely to be very different.

Smart Grid technologies use the integration of digital networks to balance the decentralised generation and consumption of power. Similarly, 'just in time' manufacturing in next generation factories enables new levels of integration between supply and demand [21]. The digital world enables operators to control massively distributed resources from a single, centralised control room. The convergence of IoT and SCADA technologies can be illustrated smart city initiatives that showcase the dynamic nature of future ICS.

Previous works that proposed adaptive intrusion detection systems [22–26] propose methods that adapt to new threats and novel attacks. Other adaptive methods that were recently propsoed [27, 28], designed mainly for MANETS, try to deal with the problems of limited transmission power, receiver collision and collaborativeattacks that may arise in such environments. None of the materials and research studied examines the adaptability or susceptibility of the proposed IDS systems to changes in the monitored architecture. This is almost certainly due to a large extent to the fact that SCADA systems tend to be relatively static. While Cheung et al [29] argue that model based IDS are suited to SCADA due to their traditionally more stable environment, this stability is decreasing as SCADA systems evolve, undermining this argument. The initial framework of an Adaptive IDS was presented by the authors earlier in [30]. In this article we further analyze the proposed A-IDS method and evaluate its performance in terms of accuracy and time overhead.

5. Dataset

In this section we describe how and why we produced our own datasets using our Hybrid testbed that we have previously validated through extensive experiments [8].

5.1. Public Datasets

Extensive research shows that the following datasets were commonly used for investigations by recognising that the KDD dataset [31], is widely known among scholars in the network analysis community, offering a large scope of records in the earlier DARPA dataset, with up to 4,900,000 training instances, 41 features, 24 training and testing attacks with a further 14 types. The dataset offers more information compared to the DARPA dataset [32]. However, Bajaj and Arora [33] state that the KDD dataset is outdated, suggesting that the NSL-KDD dataset is the most suited for current network analysis. They state that KDD 99 dataset suffers with redundant data which often lead to biased detection of attacks, highlighting more frequency in DOS and probe attack. This lead to failures in classifying features appropriately and most records cannot be classified and are misrepresented in most of the cases. The author states that if the KDD, datasets are used,investigations will likely present results that do not represent real network situations. Moreover, numerous current studies showed that for the current network threat environment, these data sets do not inclusively reflect network traffic and modern low footprint attacks. Countering the unavailability of network benchmark data set challenges, authors in [34] examined a UNSW-NB15 data set creation. This data set has a hybrid of the real modern normal and the contemporary synthesized attack activities of the network traffic. However for our current study, which tries to investigate how architectural changes affect the performance of an anomaly detection IDS, even the last dataset is not appropriate since it represents a static topology. For this reason we decided to create our own datasets, using our testbed and evaluate the performance of a static IDS and the proposed adaptive IDS.

5.2. Hybrid Testbed

The HEDVa (Hybrid Environment for Design and Validation) was developed by the Israeli Electric Company with the purpose of providing a flexible platform to support the creation and maintenance of multiple testbed environments, within a multi-tenant environment. It provides resources for component development, test and integration, which can be dynamically added and/or allocated to deployed scenarios, enabling flexible reconfiguration. The HEDVa provided the CI environment in which the development and validation of the CockpitCI concept was undertaken (see Figure 1).

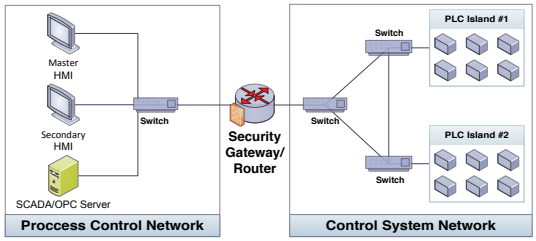

Figure 1. Simplified HEDVa networking architecture

The CockpitCI validation effort leveraged the HEDVa resources to build a hybrid CI scenario, in the sense that it makes use of real/physical SCADA and network/telecom infrastructure components to implement a simulation model of an electric grid (see Figure 2). It was developed using key performance indicators and data from production environments, also implementing standard operator Fault Detection Isolation and Recovery procedures used for electric grid management.

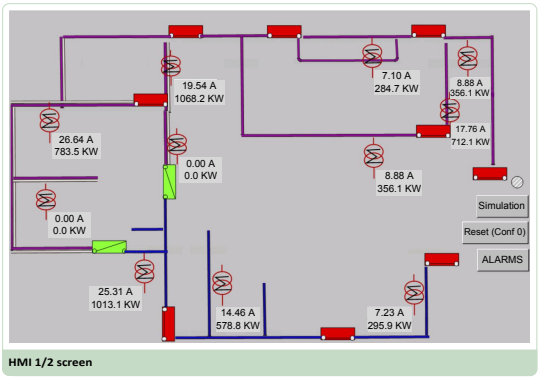

Figure 2. HEDVa grid scenario, with breakers and substation feeders

In the CockpitCI testbed, grid elements (such as feeders and breakers) are emulated by real PLC devices, which take part in the simulation - moreover, all voltage and current values on critical points are dynamically calculated and updated accordingly with the mathematical model of the power grid. By emulating the cyber-physical parameters, this scenario allows to implement several different attacks and failure use cases for validation and interdependence analysis in a safe environment, while providing a realistic attack surface.

The topology depicted in Figure 1 constitutes the CI that supports the simulated grid environment. The entire CockpitCI security detection components (in which the OCSVM IDS is included) were trained and deployed in this infrastructure, which also served as demonstrator vehicle.

Table 1. Different configurations of the network

Data	PLC Gr. 1	PLC Gr. 2	HMI 1	HMI 2
1	Active	-	Active	-
2	Active	-	Active	Active
3	Active	Active	Active	-
4	Active	Active	Active	Active
5	-	Active	Active	Active
6	-	Active	Active	-

6. Testing of OCSVM on different Architectures

Six data sets were provided using the HEDVa. These datasets represent network traffic from a SCADA network running in six different configurations. There are two groups of Programmable Logic Controllers (PLCs) and two Human Machine Interface (HMI) devices which may be active on the system (see Figure 1). Table 1 shows the configuration of the network for each of the data sets.

While the accuracy for models when tested against themselves varied very little from the mean value, the deviation for testing against other data sets was shown to be extreme. The results presented on Figure 3 show that while there are some large deviations, these have occurred primarily in data set 4, which is later removed from the study.

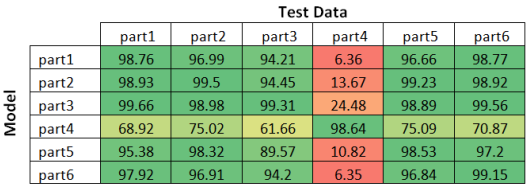

Figure 3. Initial accuracy heat map

From this we can see that the part 4 of the data set is displaying significant variation from the other sets, both when other models are tested against its data (shown by the vertical red line), and where the model for part 4 is tested against other data sets (the horizontal light green line).

To investigate the cause of this behaviour we inspected the meta data created for each PCAP file. We could see that all of the data sets exhibit a very high maximum rate value (in the region of 20 hours). We then looked at the rates meta data file created at the same time. These files contain a list of every rate value calculated when processing a PCAP file, and the number of times it occurs in that file. We found out that there existed some big gaps between transmitted packets. These gaps clearly do not represent the true traffic rate feature we are attempting to extract and could, themselves, trigger an alert from the IDS. More importantly, these

outlier values would affect the significance of the normal traffic rate values when scaling is applied. To eliminate this issue, the program was updated to filter out any rate value over 3 seconds, and to substitute the current average rate so far.

Next we investigated the IP addresses encountered in the traffic. We observed that for the majority of the data sets, all of the IPs are from the private address space 172.27.xx.xx. In the part 4 data set, however, we see that there are also 10 IP addresses from other, external networks. Another factor is the type of traffic detected in the captured data. For all sets other than part4, only TCP traffic is detected. The part4 data set, however, also shows UDP traffic as well as traffic which has not been successfully identified.

It seems clear from these indicators that the part 4 data set differs from the other data sets in a substantial way, and not merely in the architecture of the network and we decided to filter out the whole part 4 data set from the analysis. On the other hand we came to some useful conclusions about how different behavior of the system in terms of packet rate and number of sources affect the accuracy of the IDS. These findings will be used in the near future in order to create a real adaptive IDS. Eliminating the data 4 set leaves us the following results (See Figure 4.

Model \ Test Data	part1	part2	part3	part5	part6
part1	98.76	96.99	94.21	96.66	98.77
part2	98.93	99.5	94.45	99.23	98.92
part3	99.66	98.98	99.31	98.89	99.56
part5	95.38	98.32	89.57	98.53	97.2
part6	97.92	96.91	94.2	96.84	99.15

Figure 4. Final accuracy heat map

With the anomalous results from part 4 eliminated we can aggregate the data to demonstrate the combined accuracy when testing against self vs other (See Figure 5).

Model \ Test Data	Self	Other
part1	98.76	96.6575
part2	99.5	97.8825
part3	99.31	99.2725
part5	98.53	95.1175
part6	99.15	96.4675
Mean	99.05	97.0795

Figure 5. Final mean accuracy

This does demonstrate a weak overall correlation between testing against other sets and accuracy. As the values approach 100 %, any difference can be very significant, especially in an IDS where every percent of inaccuracy can represent false positives or negatives. This, in turn, can represent either missing malicious activity or generating a large number of spurious alerts which can consume monitoring resources and renders the system ineffective.

7. Proposed Adaptive IDS

In order to cope with the drop of accuracy that the IDS demonstrates when the architecture of the system changes, we propose an adaptive mechanism that can be used in any system. The mechanism that is presented on Figure 6 matches the current network traffic of the system to the traffic that the IDS was trained with. Based on the fact that the matching takes additional computation time, the proposed adaptive system imposes a delay on the performance of the IDS. The matching that is proposed on this article only takes into account the different IPs that exist on the network traffic and chooses the OCSVM that was trained to a similar network. This could be extended also to the overall traffic that exist inside the system, based on the fact that the volume of traffic changes between morning and evening and between working days and weekends, especially for a system that controls critical infrastructures that are directly related to human activity, e.g. traffic controls, smart grid e.t.c.

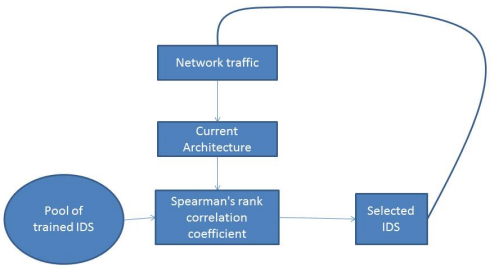

Figure 6. Proposed Adaptive IDS

In the core of our adaptive IDS we have included Spearman's rank correlation coefficient. Spearman's rank is a non-parametric measure of correlation widely used to describe the relationship between two variables that is used to report the difference in ranking produced by two methods. Based on this metric we can find the most suitable IDS for the current architecture of the network with a notion of traffic in it, since the metric ranks the

sources based on the total traffic they induce in the system.

Data: Table 1
Result: Choose the appropriate trained IDS initialization;
while *not at end of Table 1* **do**
  Read current line;
  Store to temporary table;
  if *time range of temporary table = Time Gap* **then**
    Rank existing IPs based on produced traffic;
    Compare to correspoding rankings of the pool of trained IDSs using Spearman's Rank Correlation Coefficient;
    Choose the IDS with higher outcome;
    Empty temporary table;
  else
    next
  end
end

Algorithm 1: Adaptive IDS algorithm

One important parameter of the proposed Adaptive IDS is the frequency of the comparison between the current traffic to the pool of the trained IDS and also the number of the trained IDSs that exist in order to better match the current situation of the network. These are issues that are investigated in the next subsection where the trade off between accuracy and delay is also analyzed.

8. System performance evaluation

In order to evaluate the efficiency of our proposed Adaptive IDS (A-IDS) we have performed a number of simulations using the datasets provided from HEDVa. We have tested the proposed A-IDS using different parameters as shown in table 2. We investigate how the number of different trained IDSs affect the performance of the system. In our experiments we now the number of different architectures of the system while in a general situation, an automatic method for finding the optimal set of trained IDSs would be needed. Also we investigate how frequent the comparison of current traffic to the ones used for training must be, since a small time gap would not include the necessary network traffic sample which would be necessary in order to choose the correct IDS, while on the other hand a big time gap would have as a consequence the system not to be able to follow up the topology changes fast enough.

8.1. Number of trained IDSs

In order to test the efficiency of our proposed A-IDS we have done the following arrangement. The system architecture is initially under configuration 1 and circles from one configuration to the other following this sequence: Configuration 1, Configuration 2, Configuration 3, Configuration 5 and Configuration 6 (See Figure 7. During the operation of the system both a static and the proposed dynamic A-IDS are used.

Table 2. Simulation Parameters

Parameters	Range	Default
Number of trained IDS	2 - 5	5
Time gap (min)	1 -10	5

The static IDS is initially trained using configuration 1 and the dynamic has 3 or 5 trained IDSs, each one representing a different configuration from Table 1. The system is constantly collecting the network traffic and compares it against the one that was used to train the pool of IDSs using the Spearman's rank correlation coefficient. The same experiment is conducted with the system starting from configuration 2, 3, etc. in order to cover all different configurations. Figure 8 shows the aggregated accuracy of the Static and of the dynamic A-IDS with 2,3,4 and 5 different individual trained IDSs.

It is obvious from Figure 8 that the A-IDS is more stable in terms of accuracy for all the different simulations conducted and manages to have a stable behavior while the architecture is constantly changing in a circular way between the different configurations as shown in Table 1.

8.2. Time gap

In this subsection we test how much the time gap affects the performance of the proposed A-IDS. As shown in Figure 9 when selecting a very small value the system is performing worse than the static IDS, due to the fact that most of the time the system chooses a wrong trained IDS to use. This is caused from the small number of collected packets used for choosing the matching IDS. These small datasets don't represent the actual situation inside the system, causing the A-IDS to choose most of the times wrongly. On the other hand, as shown in Figure 9, a rather big time gap leads to the degradation of the A-IDS to the static one.

8.3. Computectional Cost and Time Overhead

The complexity of an IDS can be attributed to hardware, software and operational factors. For simplicity, it is usually estimated as the computing time required to perform classification of the dataset and output the final alarms. Increasing the number of classifiers usually increases the computational cost and decreases their comprehensibility. For this reason reason, special care must be taken when choosing the number of trained IDSs. While the increase in number of trained IDSs improves the method's performance in terms of accuracy, this may slow down the detection mechanism. In Fig.

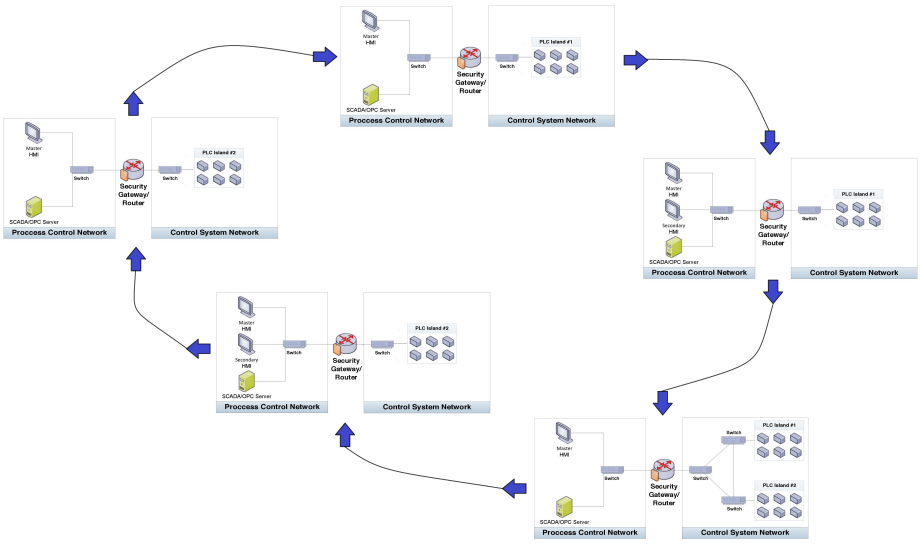

Figure 7. Topology of system circles between five different configurations

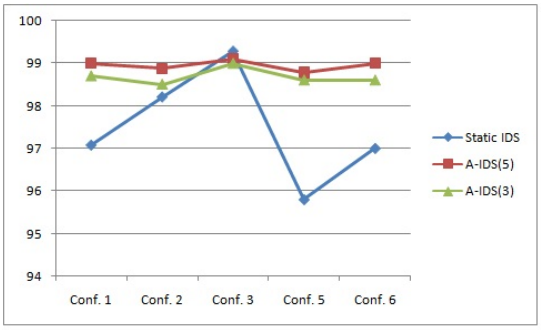

Figure 8. Aggregated accuracy of static and the proposed A-IDS

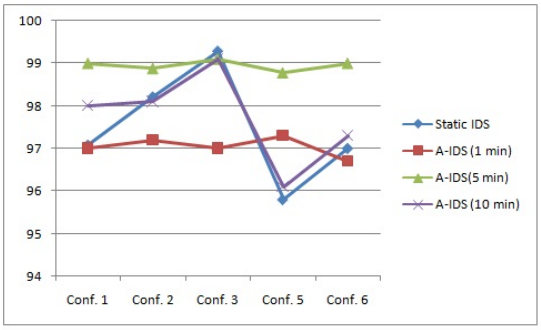

Figure 9. Time Gap affects the performance of the A-IDS

10, we illustrate the time performance of the method compared to a static IDS. The evaluation was conducted on a PC with Intel core 2 duo 1.7 MHz CPU, 2 GB main memory, 80 GB hard disk 7200 rpm hard disk and Microsoft windows 7, 64 bit.

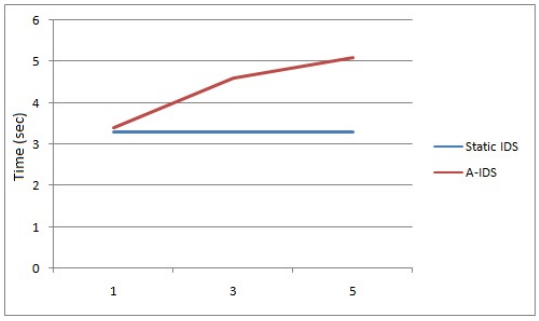

Figure 10. Approximate execution time.

According to Fig. 10, the execution time of the proposed A-OCSVM is bigger compared to a static IDS method. The performance gap is between 55% and 33% depending on the number of IDSs that were initially created. This time overhead is mainly caused by the Spearman's rank correlation coefficient method that is used in order to identify the best matching IDS. Based on these observations we conclude that the proposed A-IDS system performs a classification in a comparable time to that of a static IDS, and thus, it can be adopted in soft real-time applications.

9. Future work

The current work is a preliminary step to a full adaptive IDS. The research was conducted using specific datasets produced from our HEDV in order to investigate the efficiency of the proposed method. There exist a lot of future paths that need to be followed in order to investigate the feasibility and deploy-ability of the method.

9.1. Additional Features

A minimal set of features were used in this study which may not be representative of the features required to implement an IDS which effectively identifies malicious behaviour. It would be useful to implement additional feature extraction and generation steps and evaluate their impact on the accuracy of the A-IDS. Additional features may have a positive or a negative impact on the performance of an IDS that is based on machine learning techniques and the selection of the proper set of features is a process that requires extensive simulations [35].

9.2. Feature & Parameter Learning

Introducing an iterative machine learning stage to the method, could allow a far more optimal feature set and parameter values for the OCSVM models generation. This procedure would generate and test multiple SVM files and models using different features and parameter values. The accuracy of these would then be analysed for correlations to the accuracy of these models. In this way, it could be determined which features and parameter values are most beneficial to the accuracy, and an optimal configuration could then be chosen for the subsequent testing. Typical methods that could be used as a basis for this research are [36, 37]

9.3. Malicious Data

It would be highly beneficial to perform further testing incorporating traffic constituting malicious activity in order to more fully determine the efficacy of the IDS simulation. There are two potential methods for achieving this:

- If traffic containing known malicious data is available, then this could be integrated programmatically into the data sets. This would require some adaptation of the data during the process to ensure that time and IP values fit appropriately into the target data set.
- Malicious data could be programmatically generated for each data set. This would require a detailed analysis of the traffic involved in one or more types of attacks, and the development of code to simulate this activity in conformance with the target data set. Although this is a significant amount of work, this functionality would be highly valuable for IDS testing generally.

The need of new datasets is based on the fact that the Evaluation of IDSs using the existing benchmark data sets of KDD99 and NSLKDD does not reflect satisfactory results. This is due to three major issues, their lack of modern low footprint attack styles, their lack of modern normal traffic scenarios, and a different distribution of training and testing sets. Recently a new dataset containing malicious data was introduced and analyzed [38]. This data set has 9 types of the modern attacks fashions and new patterns of normal traffic, and it contains 49 attributes that comprise the flow based between hosts and the network packets inspection to discriminate between the observations, either normal or abnormal.

9.4. Performance Improvements

As the data sets are very large, the time taken to train and test models can become significant and impact on the overall amount of work that can be done. Further testing would benefit from any performance improvements which can reduce this time. Three potential methods for achieving greater performance have been identified. Each one would be of great benefit, but combining two or all three of these could greatly improve the throughput of the system.

Multi-threading. Multi-core processors are standard on modern computers, and using only a single core for processing can be highly inefficient. If the program could be modified to utilise multiple cores, then the performance might be improved dramatically. Although the overhead of making code thread-safe can negate the benefits, this is unlikely to be the case in this computationally heavy application. canny2013big There are two potential avenues for achieving this. The most difficult would be to modify the libsvm code to be multi-threaded. This could be very complex but would allow performance gains even when training or testing a single model. The second method would be to allow training or testing of models to be done in parallel. This would not provide a benefit when processing a single model, but is likely to be far simpler to achieve reliably. Multi-threading techniques that were recently introduced could be used as a reference to our future work citecanny2013big.

Distributed Processing. The code could be modified to operate in a client/server fashion over the network to allow the processing of models to be distributed across more than one machine. This would apply during batch operations, such as training all models, as the operations are discrete and separate from each other. In these circumstances it is simple and straightforward for another machine to perform a complete operation remotely.

GPU Processing. Graphics Processor Units (GPUs), while designed for the task of generating 3D graphics, are also excellent at performing massively parallel computations for more generic applications When exploited correctly this can produce huge speed improvements for code, with applications such as the password cracking software HashCat seeing ten time the performance with an inexpensive commercial graphics

card. Converting code to operate on GPU architecture, however there are implementations of the libsvm library which have done exactly this (although not for OCSVM), which could be used as a starting point for this development work.

10. Conclusions

The current research on IDS for SCADA systems focuses on relatively static systems. While this has been reasonable in the past when SCADA systems have remained unchanged for long periods, it is clear that as SCADA systems adopt modern technology and characteristics of IT systems they are likely to become more dynamic, leading to a need for research into how such changes affect the performance of IDSs. This article investigates how changes in the architecture of a SCADA system affect the performance of an IDS that is based on the OCSVM machine. Also the article proposes an adaptive mechanism that can cope with such changes and can work in real time situations. The proposed mechanism can be a basis for developing real time Adaptive IDS, for both IT and IC systems, that are based on classification mechanisms.

References

[1] A.-R. Sadeghi, C. Wachsmann, and M. Waidner, "Security and privacy challenges in industrial internet of things," in *Proceedings of the 52nd Annual Design Automation Conference*. ACM, 2015, p. 54.

[2] H. Kim, "Security and vulnerability of scada systems over ip-based wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, 2012.

[3] Y. Yang, K. McLaughlin, S. Sezer, T. Littler, E. G. Im, B. Pranggono, and H. Wang, "Multiattribute scada-specific intrusion detection system for power networks," *Power Delivery, IEEE Transactions on*, vol. 29, no. 3, pp. 1092–1102, 2014.

[4] L. A. Maglaras, J. Jiang, and T. Cruz, "Integrated ocsvm mechanism for intrusion detection in scada systems," *Electronics Letters*, vol. 50, no. 25, pp. 1935–1936, 2014.

[5] L. A. Maglaras, J. Jiang, and T. J. Cruz, "Combining ensemble methods and social network metrics for improving accuracy of ocsvm on intrusion detection in scada systems," *Journal of Information Security and Applications*, 2016.

[6] W. Shang, P. Zeng, M. Wan, L. Li, and P. An, "Intrusion detection algorithm based on ocsvm in industrial control system," *Security and Communication Networks*, 2015.

[7] M. Zhang, B. Xu, and J. Gong, "An anomaly detection model based on one-class svm to detect network intrusions," in *2015 11th International Conference on Mobile Ad-hoc and Sensor Networks (MSN)*. IEEE, 2015, pp. 102–107.

[8] T. Cruz, L. Rosa, J. Proença, L. Maglaras, M. Aubigny, L. Lev, J. Jiang, and P. Simões, "A cybersecurity detection framework for supervisory control and data acquisition systems," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 6, pp. 2236–2246, 2016.

[9] V. M. Igure, S. A. Laughter, and R. D. Williams, "Security issues in scada networks," *Computers & Security*, vol. 25, no. 7, pp. 498–506, 2006.

[10] A. Nicholson, S. Webber, S. Dyer, T. Patel, and H. Janicke, "Scada security in the light of cyber-warfare," *Computers & Security*, vol. 31, no. 4, pp. 418–436, 2012.

[11] T. Pandit and A. Dudy, "An artificial neural network based approach for dos attacks detection in manet," *IJERST*, 2014.

[12] Y. Wang, J. Wong, and A. Miner, "Anomaly intrusion detection using one class svm," in *Information Assurance Workshop, 2004. Proceedings from the Fifth Annual IEEE SMC*. IEEE, 2004, pp. 358–364.

[13] D. S. Kim, H.-N. Nguyen, and J. S. Park, "Genetic algorithm to improve svm based network intrusion detection system," in *Advanced Information Networking and Applications, 2005. AINA 2005. 19th International Conference on*, vol. 2. IEEE, 2005, pp. 155–158.

[14] B. M. Jahromy, A. R. Honarvar, M. Saif, and M. A. M. Jahromy, "A new method for detecting network intrusion by using a combination of genetic algorithm and support vector machine classifier," *Journal of Engineering and Applied Sciences*, vol. 100, no. 4, pp. 810–815, 2016.

[15] T. Singh, S. Verma, V. Kulshrestha, and S. Katiyar, "Intrusion detection system using genetic algorithm for cloud," in *Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies*. ACM, 2016, p. 115.

[16] L. A. Maglaras and J. Jiang, "Ocsvm model combined with k-means recursive clustering for intrusion detection in scada systems," in *Heterogeneous Networking for Quality, Reliability, Security and Robustness (QShine), 2014 10th International Conference on*. IEEE, 2014, pp. 133–134.

[17] S. Dubey and J. Dubey, "Kbb: A hybrid method for intrusion detection," in *Computer, Communication and Control (IC4), 2015 International Conference on*. IEEE, 2015, pp. 1–6.

[18] W.-C. Lin, S.-W. Ke, and C.-F. Tsai, "Cann: An intrusion detection system based on combining cluster centers and nearest neighbors," *Knowledge-based systems*, vol. 78, pp. 13–21, 2015.

[19] W. Shang, L. Li, M. Wan, and P. Zeng, "Industrial communication intrusion detection algorithm based on improved one-class svm," in *2015 World Congress on Industrial Control Systems Security (WCICSS)*. IEEE, 2015, pp. 21–25.

[20] T. Cruz, J. Proença, P. Simões, M. Aubigny, M. Ouedraogo, A. Graziano, and L. Yasakhetu, "Improving cyber-security awareness on industrial control systems: The cockpitci approach," in *13th European Conference on Cyber Warfare and Security ECCWS-2014 The University of Piraeus Piraeus, Greece*, 2014, p. 59.

[21] C. Johnson, "Securing the participation of safety-critical scada systems in the industrial internet of things," 2016.

[22] W. L. Al-Yaseen, Z. A. Othman, and M. Z. A. Nazri, "Real-time multi-agent system for an adaptive intrusion detection system," *Pattern*

Recognition Letters, vol. 85, pp. 56 – 64, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167865516303415

[23] K.-H. Lee and Y. B. Park, "A study of environment-adaptive intrusion detection system," in *International Conference on Computer Science and its Applications*. Springer, 2016, pp. 625–630.

[24] B. Mahapatra and S. Patnaik, "Self adaptive intrusion detection technique using data mining concept in an ad-hoc network," *Procedia Computer Science*, vol. 92, pp. 292–297, 2016.

[25] S. Talwar, "Data mining based classification technique for adaptive intrusion detection system using machine learning," *International Journal of Advances in Engineering Sciences*, vol. 5, no. 3, pp. 16–19, 2015.

[26] R. R. Karthick, V. P. Hattiwale, and B. Ravindran, "Adaptive network intrusion detection system using a hybrid approach," in *2012 Fourth International Conference on Communication Systems and Networks (COMSNETS 2012)*. IEEE, 2012, pp. 1–7.

[27] T. Sheltami, A. Basabaa, and E. Shakshuki, "A3acks: adaptive three acknowledgments intrusion detection system for manets," *Journal of Ambient Intelligence and Humanized Computing*, vol. 5, no. 4, pp. 611–620, 2014.

[28] A. Nadeem and M. P. Howarth, "An intrusion detection & adaptive response mechanism for manets," *Ad Hoc Networks*, vol. 13, pp. 368–380, 2014.

[29] S. Cheung, B. Dutertre, M. Fong, U. Lindqvist, K. Skinner, and A. Valdes, "Using model-based intrusion detection for scada networks," in *Proceedings of the SCADA security scientific symposium*, vol. 46. Citeseer, 2007, pp. 1–12.

[30] B. Stewart, L. Rosa, L. Maglaras, T. Cruz, P. Simões, and H. Janicke, "Effect of network architecture changes on ocsvm based intrusion detection system," in *INISCOM 2016, LNICST 188*, 2017.

[31] M. Sabhnani and G. Serpen, "Application of machine learning algorithms to kdd intrusion detection dataset within misuse detection context." in *MLMTA*, 2003, pp. 209–215.

[32] C. Thomas, V. Sharma, and N. Balakrishnan, "Usefulness of darpa dataset for intrusion detection system evaluation," in *SPIE Defense and Security Symposium*. International Society for Optics and Photonics, 2008, pp. 69 730G–69 730G.

[33] K. Bajaj and A. Arora, "Dimension reduction in intrusion detection features using discriminative machine learning approach," *International Journal of Computer Science Issues (IJCSI)*, vol. 10, no. 4, 2013.

[34] N. Moustafa and J. Slay, "Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)," in *Military Communications and Information Systems Conference (MilCIS), 2015*. IEEE, 2015, pp. 1–6.

[35] O. Maimon and L. Rokach, *Data mining and knowledge discovery handbook*. Springer, 2005, vol. 2.

[36] J. Yang, T. Deng, and R. Sui, *An Adaptive Weighted One-Class SVM for Robust Outlier Detection*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 475–484. [Online]. Available: http://dx.doi.org/10.1007/978-3-662-48386-2_49

[37] S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie, "High-dimensional and large-scale anomaly detection using a linear one-class svm with deep learning," *Pattern Recognition*, vol. 58, pp. 121–134, 2016.

[38] N. Moustafa and J. Slay, "The evaluation of network anomaly detection systems: Statistical analysis of the unsw-nb15 data set and the comparison with the kdd99 data set," *Information Security Journal: A Global Perspective*, 2016.