Security Solutions and Applied Cryptography in Smart Grid Communications

Mohamed Amine Ferrag
Guelma University, Algeria

Ahmed Ahmim
University of Larbi Tebessi, Algeria

A volume in the Advances in Information Security, Privacy, and Ethics (AISPE) Book Series

www.igi-global.com

Published in the United States of America by
IGI Global
Information Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue
Hershey PA, USA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: http://www.igi-global.com

Library of Congress Cataloging-in-Publication Data

Names: Ferrag, Mohamed Amine, 1987- editor. | Ahmim, Ahmed, 1986- editor.
Title: Security solutions and applied cryptography in smart grid communications / Mohamed Amine Ferrag and Ahmed Ahmim, editors.
Description: Hershey PA : Information Science Reference, [2017] | Includes index.
Identifiers: LCCN 2016045997| ISBN 9781522518297 (hardcover) | ISBN 9781522518303 (ebook)
Subjects: LCSH: Smart power grids--Security measures. | Data encryption (Computer science) | Computer networks--Security measures.
Classification: LCC TK3105 .S334 2017 | DDC 621.3190285/58--dc23 LC record available at https://lccn.loc.gov/2016045997

This book is published in the IGI Global book series Advances in Information Security, Privacy, and Ethics (AISPE) (ISSN: 1948-9730; eISSN: 1948-9749)

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

Chapter 9
Novel Intrusion Detection Mechanism with Low Overhead for SCADA Systems

Leandros Maglaras
De Montfort University, UK

Jianmin Jiang
Shenzhen University, China

Helge Janicke
De Montfort University, UK

Andrew Crampton
University of Huddersfield, UK

ABSTRACT

SCADA (Supervisory Control and Data Acquisition) systems are a critical part of modern national critical infrastructure (CI) systems. Due to the rapid increase of sophisticated cyber threats with exponentially destructive effects, intrusion detection systems (IDS) must systematically evolve. Specific intrusion detection systems that reassure both high accuracy, low rate of false alarms and decreased overhead on the network traffic must be designed for SCADA systems. In this book chapter we present a novel IDS, namely K-OCSVM, that combines both the capability of detecting novel attacks with high accuracy, due to its core One-Class Support Vector Machine (OCSVM) classification mechanism and the ability to effectively distinguish real alarms from possible attacks under different circumstances, due to its internal recursive k-means clustering algorithm. The effectiveness of the proposed method is evaluated through extensive simulations that are conducted using realistic datasets extracted from small and medium sized HTB SCADA testbeds.

INTRODUCTION

In order to modernize the national critical infrastructure, cyber-physical systems are becoming a vital part of them. Cyber-attacks tend to target important assets of the system, taking advantage of vulnerabilities on the architecture design or weaknesses of the defense systems. Lately several research efforts have revealed the importance of human factor on the cyber security assurance of a system (Evans, 2016; Ayres, 2016). Most of the weaknesses in CIs arise from the fact that system architects tend to adopt off-the-shelf technologies from the IT world, without a significant change, thus relying on the "airgap"

DOI: 10.4018/978-1-5225-1829-7.ch009

security principle that falsely assumes that an apparently isolated and obscure systems are implicitly secure. The integration of new technologies, especially Internet-like communications networks, may introduce some new threats to the security of a smart grid. In such a network there are three crucial aspects of security that may be threatened due to the CIA-triad, these being: confidentiality, integrity, and availability (Woo, 2015)

- Confidentiality is the property that information is not made available or disclosed to unauthorized individuals, entities or processes. An attack on this occurs when an unauthorized person, entity or process enters the system and accesses the information.
- Integrity refers to safeguarding the accuracy and completeness of assets, which ensures that the information in the system will not be modified by attacks.
- Availability pertains to the property of being accessible and usable upon demand by an authorized entity. The resources need to be kept accessible at all times to authorized entities or processes.

The integration of new technologies such as smart meters and sensors can bring new vulnerabilities to a smart grid that combined with the traditional cyber threats like malware, spyware and computer viruses make the situation complex and hard to deal with. (Sadeghi, 2015). In the three main control systems of a CI, the SCADA is the central nerve system that constantly collects the latest status from remote units, such as RTUs and PLCs. The communication between the different sub networks and the control system of a power grid can be blocked or cut off due to component failures or communication delays. If one of the crucial communication channels fails to connect in the operational environment, the control of important facilities may be impossible leading to possible power outages. In this situation, the effect of some widely known attacks can have devastating consequences on SCADA systems.

Intrusion detection systems can be classified into centralized intrusion detection systems (CIDS) and distributed intrusion detection systems (DIDS) depending on how the different components are distributed (Kenkre, 2014). In a CIDS the analysis of the data is performed in some fixed locations independently on the number of hosts that are monitored, while in a DIDS several IDS can be located in different places inside the smart grid. DIDS has specific advantages over CIDS. For instance, it is highly scalable easily extensible and scalable (Vasilomanolakis, 2014). It is evident that the development of distributed IDS specifically designed for SCADA systems, being able to ensure an adequate balance between high accuracy, low false alarm rate and reduced network traffic overhead, is needed. The above discussion clearly indicates that specific intrusion detection systems that reassure both high accuracy, low rate of false alarms and decreased overhead on the network traffic need to be designed for SCADA systems. Based on this need, new IDSs are constantly introduced belonging to two main categories; signature based and misuse detection. There has been considerable amount of work regarding SCADA intrusion and anomaly detection. Some IDS solutions involve combining network traces and physical process control data (Gao, 2010), other focus on detection of anomalies on network traffic (Yang, 2014) while there exist approaches that use machine-learning techniques (Maglaras, 2014), among others.

BACKGROUND

Intrusion detection can be categorized based on principal system characteristics as anomaly detection, signature detection and hybrid/compound detection. Countless research has been con- ducted in an

attempt to answer the question of how to study the effectiveness of intrusion detection systems and how to handle attacks against intrusion detection systems themselves (Khan, 2007; Mukkamala, 2002; Maglaras, 2016; Cook, 2016). Although many quality contributions exist in this area, there is still plenty of space to improve areas in IDS development. Based on expert opinions there is a general consensus regarding the current state of network IDS. Many organizations are opting into purchasing signature based intrusion detection systems, due to the fact that they require less supervision, offer more automation and consume less time in setting features; therefore there is a belief that chances of human error are reduced. Furthermore, its widely stated that the majority of these organisations will employ IDSs that are not suited to their system needs as they simply pick the biggest brands, which may offer simplicity but on the same time they are left without an understanding of how to use these systems. Many experts state that issues still remain in identifying new forms of intrusion and in order to stay ahead, the cyber security industry must continue to develop IDS and organisations train key staff on how to use these devices rather than relying solely on automation

Misuse Detection Systems, commonly referred to as signature based detection systems since they work by using patterns of recognized attacks or known critical points in a system, can be used in order to find and match known intrusions. One big family of such intrusion detection algorithms is rule based algorithms (Roesch, 1999). Misuse detection offers greater accuracy and can efficiently detect variations of recognized attacks. Furthermore, such IDS also offer more meaningful intrusion diagnostics when an alarm is triggered by detailing diagnostic information about the cause of an alarm. In real applications though, during abnormal situations, the behavior of the system cannot be predicted and does not follow any known pattern or rule. This characteristic makes rule based algorithms incapable of detecting novel intrusions.

An anomaly detection based system uses the normal profile of a system or user to determine its decision making process, (Ahmed, 2016; Maglaras L. A., 2014; Shang, 2015). Development begins at the point at which the detector forms a judgment on behavior that constitutes to normal for the observed object in question and then a percentage of this activity may be flagged as suspicious and a preserved action is then taken. Generally, anomaly detection can be regarded as binary classification problem and thus many classification algorithms which are utilized for detecting anomalies, such as neural networks, support vector machines, K-nearest neighbor (KNN) and Hidden Markov model can be used. However, strictly speaking, they are not intrusion detection algorithms, as they require knowing what kind of anomaly is expecting, which deviates the fundamental object of intrusion detection. In addition these algorithms may be sensitive to noise in the training samples.

Segmentation and clustering algorithms (Portnoy, 2001) seem to be better choices because they do not need to know the signatures of the series. The shortages of such algorithms are that they always need parameters to specify a proper number of segmentation or clusters and the detection procedure has to shift from one state to another state. Negative selection algorithms (Kim, 2001) on the other hand, are designed for one-class classification; however, these algorithms can potentially fail with the increasing diversity of normal set and they are not meant to the problem with a small number of self-samples, or general classification problem where probability distribution plays a crucial role. Furthermore, negative selection only works for a standard sequence, which is not suitable for on line detection. Other algorithms, such as time series analysis (Viinikka, 2006) are also used as anomaly detection systems, but again, they may not be suitable for most of the real application cases.

To minimize the above mention drawbacks an intelligent approach based on OCSVM [One- Class Support Vector Machine] principles is proposed for intrusion detection. OCSVM is a natural extension

of the support vector algorithm to the case of unlabeled data, especially for detection of outlier. The OCSVM algorithm maps input data into a high dimensional feature space (via a kernel) and iteratively finds the maximal margin hyperplane which best separates the training data from the origin.

OCSVM principles have shown great potential in the area of anomaly detection (Wang, 2004; Ma, 2003; Li, 2003). IDS can provide active detection and automated responses during intrusions (Dasgupta, 2001). The CockpitCI Framework detailed in (Cruz, 2014) 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 Commercial IDS products such as NetRanger, RealSecure, and Omniguard Intruder alert work on attack signatures. These signatures needed to be updated by the vendors on a regular basis in order to protect from new types of attacks. Most of the current intrusion detection commercial software are based on approaches with statistics embedded feature processing, time series analysis and pattern recognition techniques. Several extensions of OCSVM method have been introduced lately (Glazer, 2013), (Song, 2008). OCSVM similar to other one-class classifiers e.g. GDE (Eskin, 2002), PGA (Knorr, 1997), suffer from false positive and over fitting situations. Intrusion detection systems (IDS) fail to deal with all kinds of attacks, while on the other hand, false alarms that are arisen from high sensitive IDS arise high economic risks.

For the OCSVM with an RBF kernel, two parameters σ and ν need to be carefully selected in order to obtain the optimal classification result. A common strategy is to separate the data set into two parts, of which one is considered unknown. The prediction accuracy obtained from the unknown set more precisely reflects the performance on classifying an independent data set. An improved version of this procedure is known as cross-validation. Cross-validation is a model validation technique for assessing how the results of a statistical analysis will generalize to an independent data set. It is mainly used in settings where the goal is prediction, and one wants to estimate how accurately a predictive model will perform in practice.

In ν-fold cross-validation (Burman, 1989; Friedman, 2001), the training set is divided into ν subsets of equal size. Sequentially one subset is tested using the classifier trained on the remaining ν-1 subsets. Thus, each instance of the whole training set is predicted once so the cross-validation accuracy is the percentage of data which are correctly classified. The cross-validation procedure can prevent the over fitting problem.

Using an ensemble of decision mechanisms with different parameters is another method to have an optimal result. An ensemble of classifiers (Menahem, 2013) is a set of classifiers whose individual decisions are combined in some way. more trusted final decision. Ensemble systems of classifiers are widely used for intrusion detection in networks. Classifier ensemble design aims to include mutually complementary individual classifiers which are characterized by high diversity either in terms of classifier structure (Tsoumakas, 2004), internal parameters (Kim M. J., 2010) or classifier inputs (Krawczyk, 2014).

Unnthorsson (2003) proposed another method to select parameters for the OCSVM. In their method, ν was first set to a user-specified allowable fraction of misclassification of the target class (e.g. 1% or 5%), then the appropriate σ value was selected as the value for the classification accuracy curve of training samples first reaches $1 - \nu$. The obtained ν and σ combination can then be used in the OCSVM classification.

OCSVM similar to other one-class classifiers suffer from false positive and over fitting. The former is a situation that occurs when the classifier fires an alarm in the absence of real anomaly in the system

and happens when parameter σ has too large vale. The latter is the situation when a model begins to memorize training data rather than learning to generalize from trend and it shows up when parameter σ is given relatively small value (Li X. L., 2008).

In this book chapter we present a novel method that is based on the combination of OCSVM method with a recursive k-means clustering (Maglaras L. A., 2014, 2015, 2014) separating the real from false alarms in real time and with no pre-selection of parameters σ and ν. The proposed method is a natural extension of the support vector algorithm to the case of unlabeled data, especially for detection of outliers. The novel *K−OCSVM* mechanism is trained offline by network traces, after the attributes are extracted from the network dataset. Output of the detection module is communicated to the system by IDMEF files that contain information about the source, time and severity of the intrusion. After the execution of the *K−OCSVM* method only severe alerts are communicated to the system by IDMEF files that contain information about the source, destination, protocol and time of the intrusion. The main feature of *K−OCSVM* module is that it can perform anomaly detection in a time-efficient way, with good accuracy and low overhead.

OCSVM METHOD

The one-class classification problem is a special case of the conventional two-class classification problem, where only data from one specific class are available and well represented. This class is called the target class. Another class, which is called the outlier class, can be sampled very sparsely, or even not at all. This smaller class contains data that appear when the operation of the system varies from the normal, due to a possible attack. In general cases, the outlier class might be very difficult or expensive to measure. Therefore, in the one class classifier training process, mainly samples from the target class are used and there is no information about its counterpart. The boundary between the two classes has to be estimated from data in the only available target class. Thus, the task is to define a boundary around the target class, such that it encircles as many target examples as possible and minimizes the chance of accepting outliers.

Scholkopf (2007) developed an OCSVM algorithm to deal with the one-class classification problem. The OCSVM may be viewed as a regular two-class SVM, where all the training data lie in the first class, and the origin is taken as the only member of the second class. The OCSVM algorithm first maps input data into a high dimensional feature space via a kernel function and then iteratively finds the maximal margin hyperplane, which best separates the training data from the origin. Using the kernel function to project input vectors into a feature space, nonlinear decision boundaries are allowed. Generally, four types of kernel are often used: linear, polynomial, sigmoid and Gaussian radial basis function (RBF) kernels.

Although the OCSVM requires samples of the target class only as training samples, some studies showed that when negative examples (i.e. samples of outlier classes) are available, they can be used during the training to improve the performance of the OCSVM. In this paper only normal data were used for the training of the method, though a similar to the one proposed by Tax (Tax, 2001), which includes a small amount of samples of the outlier class, will be also applied and evaluated in the near future.

OCSVM FOR SCADA SYSTEM

Cyber-attacks against SCADA systems (Barbosa, 2010) are considered extremely dangerous for Critical Infrastructure (CI) operation and must be addressed in a specific way (Zhu, 2011). Presently one of the most adopted attacks to a SCADA system is based on fake commands sent from the SCADA to the RTUs. OCSVM (Jiang, 2013; Zhang, 2008) possesses several advantages for processing SCADA environment data and automate SCADA performance monitoring, which can be highlighted as:

- In the case of SCADA performance monitoring, which patterns in data are normal or abnormal may not be obvious to operators. Since OCSVM does not require any signatures of data to build the detection model it is well suited for intrusion detection in SCADA environment.
- Since the detection mechanism does not require any prior information of the expected attack types, OCSVM is capable of detection both known and unknown (novel) attacks.
- In practice training data, taken from SCADA environment, could include noise samples. Most of the classification based intrusion detection methods are very sensitive to noise. However, OCSVM detection approach is robust to noise samples in the training process.
- Algorithm configuration can be controlled by the user to regulate the percentage of anomalies expected.
- Due to the low computation time, OCSVM detectors can operate fast enough for online performance monitoring of SCADA systems.
- Typically monitoring data of SCADA systems consists of several attributes and OCSVM is capable of handling multiple attributed data.

K−OCSVM

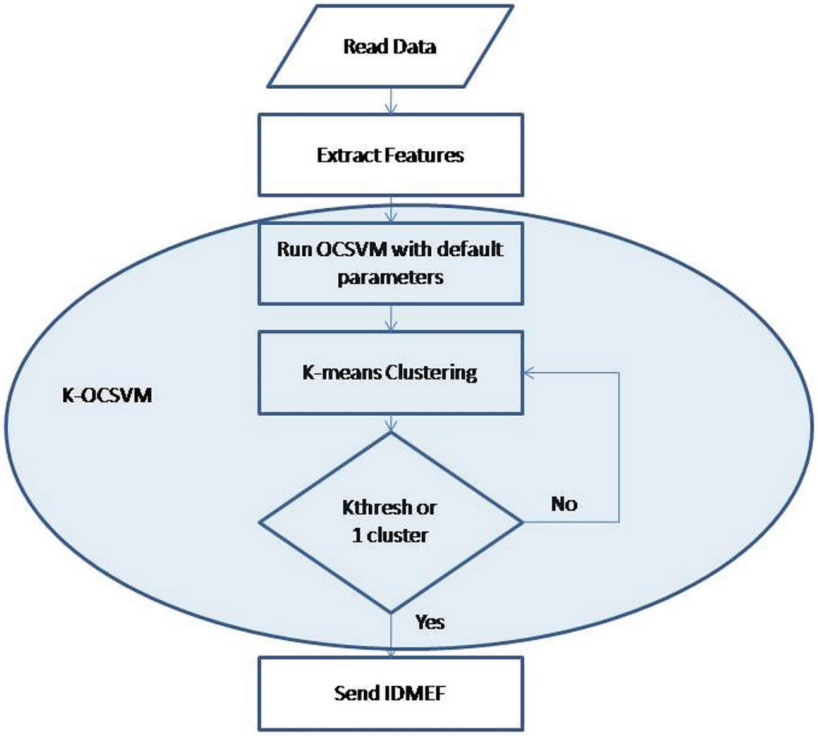
The proposed *K−OCSVM* combines the well-known OCSVM classifier with the RBF kernel with a recursive K-means clustering module. Figure 1 illustrates the procedure of intrusion detection of our proposed K−*OCSVM model*.

The OCSVM classifier runs with default parameters and the outcome consists of all possible outliers. These outliers are clustered using the k-means clustering method with *2* clusters, where the initial means of the clusters are the maximum and the minimum negative values returned by the OCSVM module. From the two clusters that are created from the K-means clustering, the one that is closer to the maximum negative value (severe alerts) is used as input in the next call of the K-means clustering. This procedure is repeated until all outcomes are put in the same cluster or the divided set is big enough compared to the initial one, according to the threshold parameter k_{thres}.

K-means clustering method divides the outcomes according to their values and those outcomes with most negative values are kept. That way, after the completion of this recursive procedure only the most severe alerts are communicated from the K−OCSVM. The division of the data need no previous knowledge about the values of the outcomes which may vary from -0.1 to -160 depending of the assigned values to parameters σ and ν. The method can find the most important/possible outliers for any given values to parameters σ and ν.

One important parameter that affects the performance of K−OCSVM is the value of threshold kthres. For given value 2, the final cluster of severe alerts that the method communicates to other parts of the IDS system is limited to 2 to 4 alarms. For bigger value (3 or more) the number of alerts also rises till

Figure 1. K−OCSVM module



the method degrades to the initial OCSVM. The optimal value for the given parameter kthres is a matter for future investigation. In order to cooperate with the other modules the OCSVM module needed to be integrated in the *PID* system and communicate with the other modules. Once an intrusion is detected several actions can be taken by the *IDS* (intrusion detection system. These actions include recording of intrusions in log files, sending of alert messages, limit the bandwidth of the intruder or even block all connections from the intruder. In order to better cooperate with the other components/modules that are being used in parallel as detection agents, the OCSVM model sends IDMEF files.

PERFORMANCE EVALUATION

Training of OCSVM Model

The initial training of both the OCSVM and the *K−OCSVM* modules is conducted using several trace files

- A trace file that is sniffed out of a typical wireless network that consists of 10.000 lines each representing a packet send in the network.
- Datasets of a testbed under normal operation
- Datasets of a medium sized SCADA system under normal operation

To train the OCSVM, we adopt the *RBF* for the kernel equation. This kernel nonlinearly maps samples into a higher dimensional space so it can handle the case when the relation between class labels and attributes is nonlinear.

The training model that is extracted after the training of the OCSVM is used for on line detection of malicious data. Since the model is based on features that are related to network traffic, and since the traffic of the system varies from area to area and from time period to time period, possible generation of multiple models could improve the performance of the module.

The network traffic in electric grids varies according to the activity which is not constant during the day. Also in some areas the activity follows different patterns according to the local demand. These characteristics maybe be critical for the proper training of the module and the accurate detection of intruders.

Testing of OCSVM Model

We evaluate the performance of the method using data from the wireless network of the University campus, from a testbed that mimics a small-scale SCADA system and from a Hybrid testbed of a medium sized SCADA system. The parameters used for the evaluation of the performance of *K−OCSVM* are listed in Table 1.

Wireless Network

In order to test our model we use another network trace file sniffed from the wireless network. The testing trace file consists of it 30.000 lines. We compare the performance of our proposed model against OCSVM classifiers having the same values for parameters σ and ν. We name each OCSVM classifier according to the parameters σ and ν: OCSV M0.07,0.01 stands for OCSVM classifier with parameters $\sigma = 0.07$ and $\nu = 0.01$.

In Table 2 we show the number of observed anomalies detected from OCSVM and *K−OCSVM* respectively. From this table it is shown how parameters σ, ν affect the performance of OCSVM. Even for a value of ν equal to *0.005*, OCSVM produces over *400* possible attacks, making the method inappropriate for a SCADA system where each false alarm is costly.

In Figure 2 and Figure 3 we present the outcome that OCSVM produces for the training network trace under different values of parameters σ and ν. From these figures it is obvious that the outcome is strongly affected by the values of these parameters, making *K−OCSVM* necessary tool for proper intrusion detection.

Table 1. Evaluation parameters

Parameter	Range of Values	Default value
σ	0.1 - 0.0001	0.007
ν	0.002 - 0.05	0.01
k_{thresh}	2 - 3	2

Table 2. Number of detected attacks for the wireless dataset (Kthres =2)

Parameter σ	Parameter ν	K−OCSVM	OCSVM
0.007	0.002	3	408
0.007	0.01	3	299
0.007	0.005	2	408
0.0001	0.01	3	274
0.1	0.01	2	295

Figure 2. OCSVM classification outcome (parameters: σ =0.007, ν=0.001)

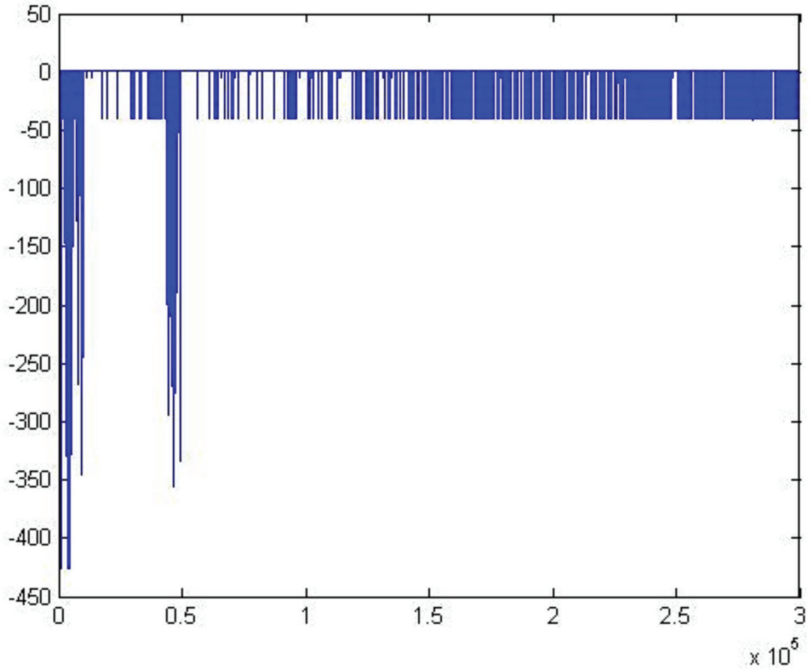
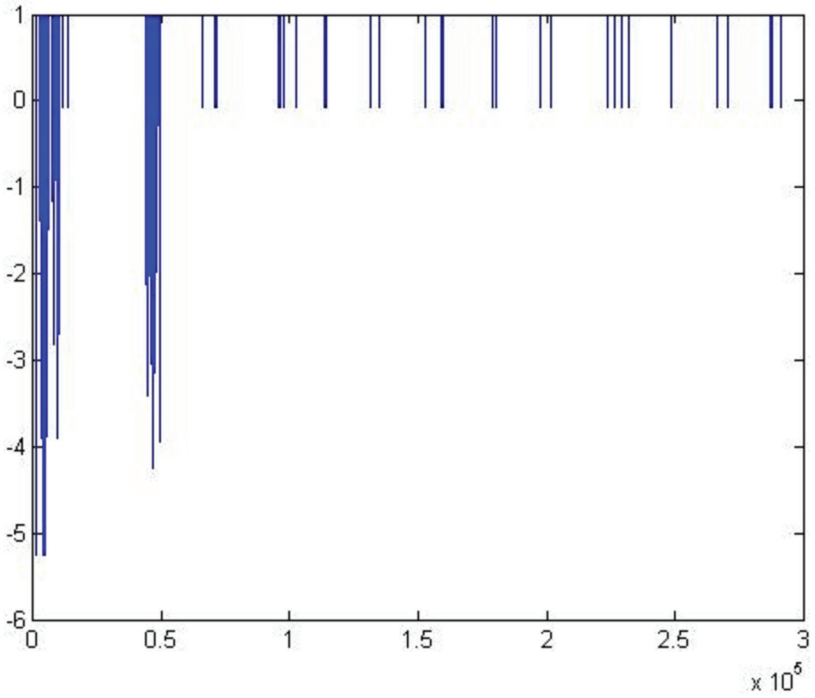


Figure 3. OCSVM classification outcome (parameters: σ =0.01, ν=0.05)



Testbed Scenario

The second trial is conducted off line with the use of two datasets extracted from the testbed. The testbed architecture mimics a small-scale SCADA system, comprising the operations and field networks and including a Human-Machine Interface Station (for process monitoring), a managed switch (with port monitoring capabilities, for net- work traffic capture), and two Programmable Logic Controller Units, for process control. The NIDS and OCSVM modules are co-located on the same host, being able to intercept all the traffic flowing on the network scopes. During the testing period several attack scenarios are simulated in the testbed. These scenarios include network scan, network flood and MITM attack.

Three kinds of attacks are being evaluated:

- **Network Scan Attack:** In typical network scan attack, the attacker uses TCP/FIN scan to determine if ports are closed to the target machine. Closed ports answer with RST packets while open ports discard the FIN message. FIN packets blend with background noise on a link and are hard to be detected.
- **ARP Cache Spoofing - MITM Attack ARP Cache Spoofing:** A technique where an attacker sends fake ARP messages. The aim is to associate the attacker's MAC address with the IP address of another host, causing any traffic meant for that IP to be sent to attacker instead. The attacker could choose to inspect the packets, modify data before forwarding (man-in-the-middle attack) or launch a denial of service attack by causing some of the packets to be dropped.
- **DoS Attack:** Network flood is the instance where the attacker floods the connection with the PLC by sending SYN packets. In a TCP SYN flooding attack, an attacker sends many SYN messages, with fictitious (spoofed) IP addresses, to a single node (victim). Although the node replies with SYN/ACK messages, these messages are never acknowledged by the client. As a result, many half open connections exist on the victim, consuming its resources. This continues until the victim has consumed all its resources, hence can no longer accept new TCP connection requests.

In Table 3 we show the number of alert messages (IDMEF) sent from OCSVM and K−OCSVM respectively. From this table it is shown how parameters σ, ν affect the performance of OCSVM for the testbed scenario. While for the same network trace file OCSVM produces from 10529 to *10704* alert messages according to the values of the parameters, *K−OCSVM* produces the same *120* alert messages. All the reported attacks are concerning the *DoS* attack that creates the biggest fluctuation in the network traffic.

Table 3. Number of produced IDMEF messages for the testbed scenario (Kthers = 2)

Parameter σ	Parameter ν	K−OCSVM	OCSVM
0.007	0.002	120	10529
0.007	0.01	120	10703
0.007	0.005	120	10584
0.0001	0.01	120	10602
0.1	0.01	120	10704

1. **Testbed Scenario with Split Testing Periods:** Since the attacks are performed during different time periods we divide the testing dataset in several smaller ones, each containing a different attack. Testing data consists of normal data and attack data and the composition of the data sets are as follows:
   a. Testing set-A': *1 - 5000*: Normal data
   b. Testing set-B': *5000 - 10000*: Normal data + **Arp spoofing** attack + **Network scan**
   c. Testing set-C': *10000 - 25000*: Normal data + **Flooding Dos attack** + **Network scan**
   d. Testing set-D': *25000 - 41000*: Normal data + **MITM attack**

From Table 4 we observe that not only the most important intrusions are detected and reported but also the total overhead on the system is limited. For all time periods the messages communicated reflect actual attacks in the network, except from the testing set-A'. In this time period *HMI* station demonstrated a significant variation in the rate that it injected packets in the system between testing and training of the module. This is due to the limited training of the OCSVM and can be avoided if training dataset consists of data that represent the traffic in the network during under workloads. The increased number of alarms created from *K−OCSVM* for the dataset B' is due to the fact in this time period the attacker uses an excessive number of SYN packets in order to flood the communication channel.

2. **Hybrid Testbed Scenario:** The third trial is conducted off line with the use of large datasets extracted from a Hybrid Testbed (*HT B*) scenario. The Hybrid testbed architecture mimics a medium-scale SCADA system, comprising the operations and field networks and including Human-Machine Interface Stations (for process monitoring), six managed switches (with port monitoring capabilities, for network traffic capture), and several Programmable Logic Controller Units, for process control. The initial dataset consists of over *3* million rows, each representing a packet sent in the system, capturing network of several days. The dataset is split in *65* smaller ones of *50.000* rows. The datasets contain only data from a normal operation of the *HTB*.

Both OCSVM and *K−OCSVM* are trained and tested with these datasets, using cross validation. The mean number of alert messages sent by the two modules is shown in Table V.

Using real datasets of a medium sized HTB SCADA system the performance of the proposed *K−OCSVM* method is very stable compared to a simple OCSVM under the same configuration. This behavior is very promising since *K−OCSVM* method has a very low false alarm rate (*lower than 0.02%*) while on the same time the overhead induced by the method is negligible (C.F. Subsection V-C. We must state

Table 4. Aggregated alarms produced by the K-ocsvm mechanism, compared to the initial alarms

Dataset	Initial alarms	Aggregated alarms
A	129	2
B	658	3
C	9273	120
D	203	3
All	10507	120

Table 5. Number of produced IDMEF messages for the Hybrid Testbed scenario (Kthers = 2)

Parameter σ	Parameter ν	K−OCSVM	OCSVM
0.007	0.002	1 - 2	40
0.007	0.01	1 - 2	207
0.007	0.005	1 - 2	105
0.0001	0.01	1 - 2	85
0.1	0.01	1 - 2	271

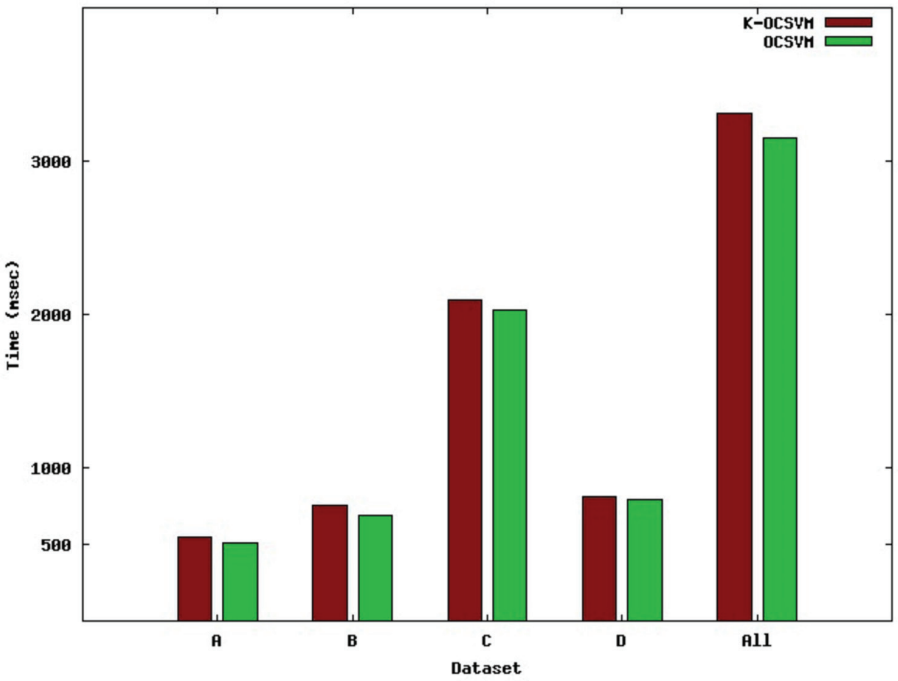
here that for the *HT B* we had available only non-malicious datasets for the evaluatioin of the proposed method. In the future when datasets containing malicious attacks are available an extensive evaluation of the *K−OCSVM* method is going to be conducted in terms of accuracy and false alarm rate.

Computational Cost and Time Overhead

Complexity of an intrusion detection system can be attributed to hardware, software and operation factors. For simplicity, it is usually estimated as the computing time required for performing classification of the dataset and outputting the final alarms. In order to evaluate the complexity of the proposed method we calculate the execution time and compare it to a simple OCSVM module. The evaluation is 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. In Figure 4 we represent the time performance of the method compared to a simple OCSVM module for the testbed scenario.

According to Figure 4 execution time of the proposed *K−OCSVM* is slightly bigger compared to a simple OCSVM method. The performance gap is around *5%* to *10%* for all the datasetsused in the simulation. Based on these observations we conclude that the system, performs a classiffication in a comparable time to that of a simple OCSVM classifier, and it thus can be adopted in soft real-time applications. We have to mention that the performance evaluation which is conducted in this subsection, does not include the time that each detection mechanism needs in order to create and disseminate IDMEF messages. It is evident that the OCSVM classifer, compared to the proposed *K−OCSVM*, needs significant additional time in order to send all the detected alarms.

Figure 4. Computational cost for the testbed scenario



FUTURE RESEARCH DIRECTIONS

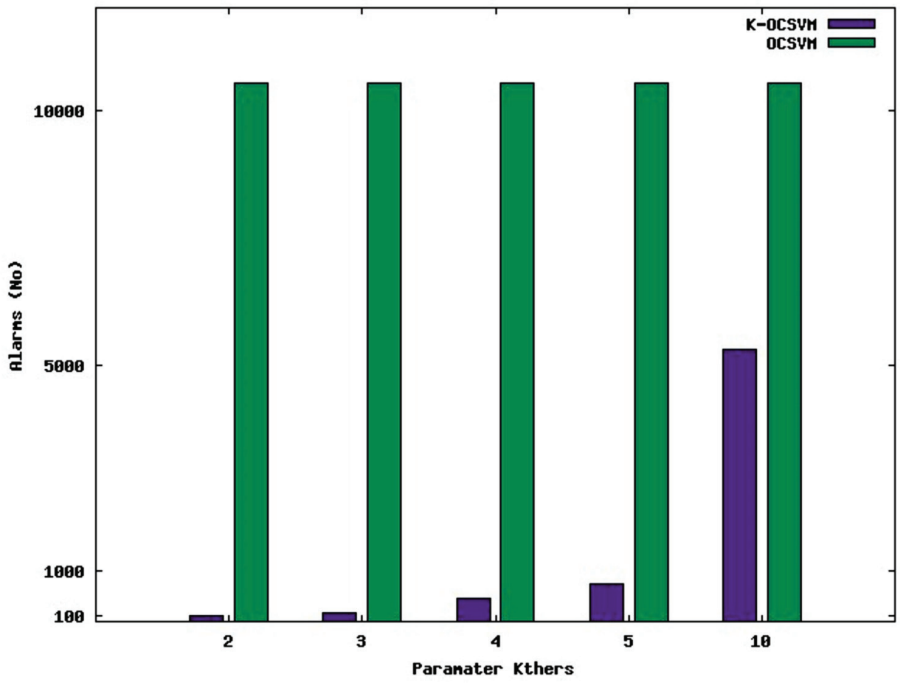
The proposed *K−OCSVM* can significantly reduce the produced alarms from the OCSVM module that is the heart of the detection mechanism. The profound advantages of low overhead and low false alarm rate come with the cost of lower accuracy and higher computational overhead. In this section we discuss some enhancements of the proposed method that can improve its performance.

Parameter k_{thres}

As stated is Section IV *K−OCSVM* method is a recursive clustering of the alarms produced by the OC-SVM module. This recursive procedure is used in order to distinguish severe from possible alarms and finally disseminate only those that represent an actual misbehavior of the system. This way the OCSVM module is enhanced in both the decreased overhead that induces to the system from the disseminated alarm files and in the decreased false alarm rate that it has. The recursive method stops when either all initial alarms are put in the same cluster or when the divided set is big enough compared to the initial one, according to the threshold parameter $k_{thre.}$

In the simulations presented in Subsection V the parameter is set to *2*. When raising the value of this parameter the produced final alarms of the proposed *K−OCSVM* method raises (See Figure 5). This raise also leads to a raise in the false alarm rate. On the other hand the accuracy of the method raises since less profound attacks are detected. By raising the value of the parameter above one limit the method degrades to the initial OCSVM. The optimum value for parameter *K−OCSVM* varies according to the architecture

Figure 5. Parameter kthres affects the performance of the K−OCSVM mechanism (Testbed scenario with default parameters σ and ν)



of the network. For big disperse networks large value of the parameter would lead to the creation of too many alarms from the module, while on the same time in a medium sized network very small value of the parameter would lead to a dangerous decrease of the detection capabilities of the module.

Except from the static configuration of the network, traffic conditions can also affect the performance of the method. In real SCADA systems the network traffic varies between daytime and night, weekdays and weekends. In order to cope with both static and dynamic features of the network an enhanced K−OCSVM method that dynamically adapts the parameter kthres, similar to (Campbell, 1999), (Cao, 2002) could be effective.

Multi Stage K−OCSVM

The proposed method uses only the values of the initial alarms in order to filter out those that don't represent an actual attack. That way attacks that cause significant variation in certain features of the OCSVM module are detected, while on the same time other more insidious attacks are passing undetected. A multi stage $K-OCSVM$ where both the number of attacks that share common characteristics, like origin, destination, port number e.t.c., and the actual values of the attacks can be developed in order to better detect different kinds of attacks.

Figure 6 represents a possible architecture of a two stage $K-OCSVM$ module. The number of the stages can be increased and the alarms produced by each stage can be further aggregated. The fusion of the outputs of the different stages can be done using any of the existing ensemble methods.i.e. majority voting, performance weighting, distribution summation, order statistics e.t.c.

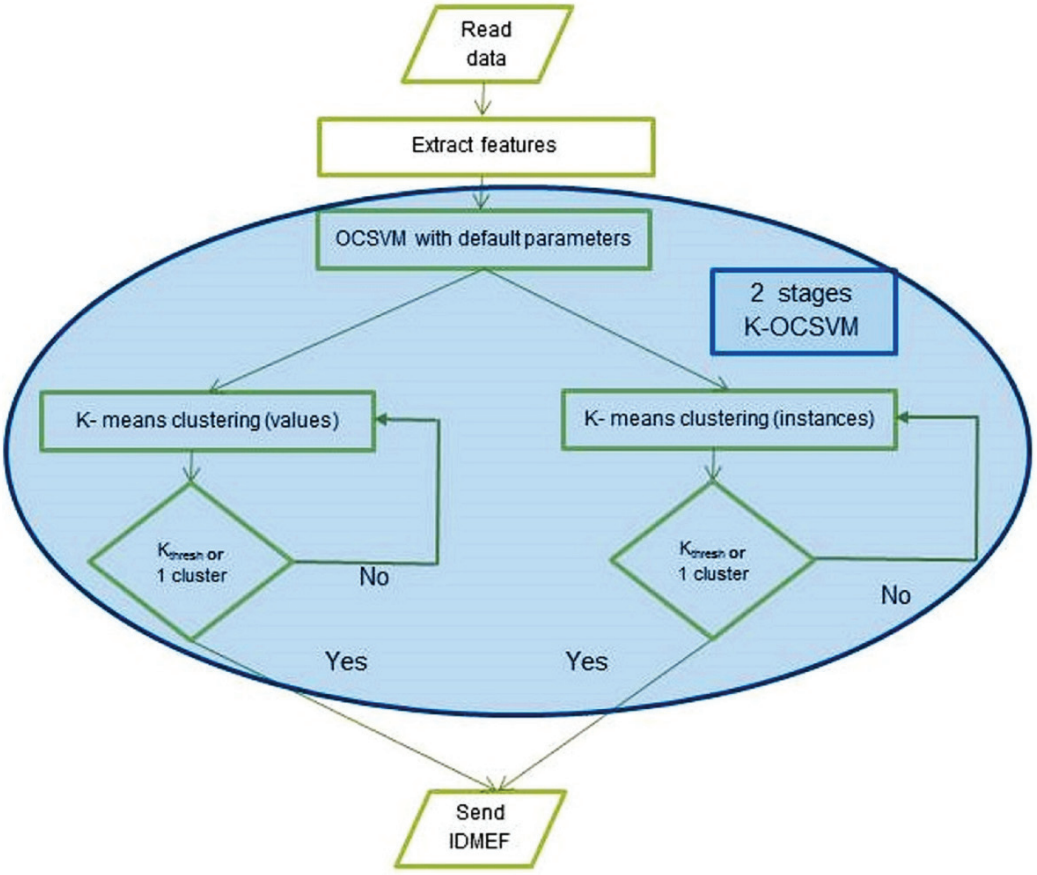
The proposed $K-OCSVM$ may not be sufficient to build effective IDS on its own but is highly valuable when coupled with other methods, especially as an integrated part of a larger framework. It is likely that while anomaly detection provides opportunity to detect unknown attacks it will be necessary to combine them with signature and ruled based IDS components to achieve the greatest accuracy. Yang et al (Yang, 2014) have shown that multiple techniques can be used to create a highly effective IDS and Maglaras (Maglaras L. A., 2014) argue that model based systems alone are insufficient. Wang (Wang, 2004) crucially note that a greater understanding of the range of SCADA applications and protocols is required to achieve truly effective model based IDS components.

CONCLUSION

We have presents an intrusion detection module for SCADA systems that is based in OCSVM classifier and a recursive k-means clustering method. The module is trained off-line by network traces, after the attributes are extracted from the network dataset. The intrusion detection module is part of an distributed IDS system.

The method is tested on three different datasets. For the first testing scenario, traces of a wireless network are used. This test shows that the method is stable and its performance is not influenced by the selection of parameters ν and σ. For the second scenario, testing of the proposed module is conducted with datasets that are sniffed of a testbed that mimics a small-scale SCADA system under different attack scenarios. After the completion of the test, not only the most important intrusions are detected and reported by $K-OCSVM$ but also the total overhead on the system is limited. Finally extensive testing of the $K-OCSVM$ module with real datasets extracted from a medium sized HTB SCADA system shows

Figure 6. Architecture of a Multi stage K−OCSVM



that the performance of the proposed *K−OCSVM* method remains very stable under different configurations. After the execution of the *K−OCSVM* method, for all the simulated scenarios, only severe alerts are communicated to the system by IDMEF files that contain information about the source, destination, protocol and time of the intrusion.

The main feature of *K−OCSVM* module is that it can perform anomaly detection in a time- efficient way, with good accuracy and low overhead. Low overhead is an important evaluation metric of a distributed detection module that is scattered in a real-time system, since frequent communication of IDMEF files from detection agents degrade the performance of the SCADA network. Recursive k-means clustering, reassures that small fluctuations on network traffic, which most of the times cause OCSVM to trigger false alarms, are ignored by the proposed detection module. The added computational time of the method compared to a simple OCSVM varies between 5% and 10% which results in a neglective time overhead. This overhead does not include the time that each detection mechanism needs in order to create and disseminate IDMEF messages. By adding the time needed in order to create and send each IDMEF file to the IDS management system the proposed *K−OCSVM* method prevails on the overall performance in terms of time efficiency. Finally we investigate how parameter k_{thres} affects the performance of the method and proposed a multi-stage *K−OCSVM* method for better accuracy.

REFERENCES

Ahmed, M., Mahmood, A. N., & Hu, J. (2016). A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, *60*, 19–31. doi:10.1016/j.jnca.2015.11.016

Ayres, N., & Maglaras, L. A. (2016). Cyberterrorism targeting the general public through social media. Security Comm. *Networks*. doi:10.1002/sec.1568

Barbosa, R. R. R., & Pras, A. (2010, June). Intrusion detection in SCADA networks. In *IFIP International Conference on Autonomous Infrastructure, Management and Security* (pp. 163-166). Springer Berlin Heidelberg.

Burman, P. (1989). A comparative study of ordinary cross-validation, v-fold cross-validation and the repeated learning-testing methods. *Biometrika*, *76*(3), 503–514. doi:10.1093/biomet/76.3.503

Campbell, C., Cristianini, N., & Shawe-Taylor, J. (1999). Dynamically adapting kernels in support vector machines. *Advances in Neural Information Processing Systems*, *11*, 204–210.

Cao, L., & Gu, Q. (2002). Dynamic support vector machines for non-stationary time series forecasting. *Intelligent Data Analysis*, *6*(1), 67–83.

Cook, A., Nicholson, A., Janicke, H., Maglaras, L., & Smith, R (2016). Attribution of Cyber Attacks on Industrial Control System. *EAI Transactions on Industrial Networks and Intelligent Systems*, 1-15.

Cruz Cruz, T., Proença, J., Simões, P., Aubigny, M., Ouedraogo, M., Graziano, A., & Yasakhetu, L. (2014, July). Improving cyber-security awareness on industrial control systems: The CockpitCI approach. In *13th European Conference on Cyber Warfare and Security ECCWS-2014*(p. 59).

Dasgupta, D., & Gonzalez, F. A. (2001, May). An intelligent decision support system for intrusion detection and response. In *International Workshop on Mathematical Methods, Models, and Architectures for Network Security* (pp. 1-14). Springer Berlin Heidelberg. doi:10.1007/3-540-45116-1_1

Eskin, E., Arnold, A., Prerau, M., Portnoy, L., & Stolfo, S. (2002). A geometric framework for unsupervised anomaly detection. In Applications of data mining in computer security (pp. 77-101). Springer US. doi:10.1007/978-1-4615-0953-0_4

Evans, M., Maglaras, L. A., He, Y., & Janicke, H. (2016). Human Behaviour as an aspect of Cyber Security Assurance. *arXiv preprint arXiv:1601.03921*

Friedman, J., Hastie, T., & Tibshirani, R. (2001). The elements of statistical learning (Vol. 1). Springer.

Gao, W., Morris, T., Reaves, B., & Richey, D. (2010, October). On SCADA control system command and response injection and intrusion detection. *IneCrime Researchers Summit*, *2010*, 1–9.

Glazer, A., Lindenbaum, M., & Markovitch, S. (2013). q-ocsvm: A q-quantile estimator for high-dimensional distributions. In Advances in Neural Information Processing Systems (pp. 503-511).

Jiang, J., & Yasakethu, L. (2013, October). Anomaly detection via one class svm for protection of scada systems. In *Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2013 International Conference on* (pp. 82-88). IEEE. doi:10.1109/CyberC.2013.22

Kenkre, P. S., Pai, A., & Colaco, L. (2015). Real time intrusion detection and prevention system. In *Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2014* (pp. 405-411). Springer International Publishing. doi:10.1007/978-3-319-11933-5_44

Khan, L., Awad, M., & Thuraisingham, B. (2007). A new intrusion detection system using support vector machines and hierarchical clustering. *The VLDB Journal—The International Journal on Very Large Data Bases, 16*(4), 507-521.

Kim, J., & Bentley, P. J. (2001, July). An evaluation of negative selection in an artificial immune system for network intrusion detection. In *Proceedings of GECCO* (pp. 1330-1337).

Kim, M. J., & Kang, D. K. (2010). Ensemble with neural networks for bankruptcy prediction. *Expert Systems with Applications*, *37*(4), 3373–3379. doi:10.1016/j.eswa.2009.10.012

Knorr, E. M., & Ng, R. T. (1997, August). *A Unified Notion of Outliers: Properties and Computation* (pp. 219–222). KDD.

Krawczyk, B., & Woźniak, M. (2014). Diversity measures for one-class classifier ensembles. *Neurocomputing*, *126*, 36–44. doi:10.1016/j.neucom.2013.01.053

Li, K. L., Huang, H. K., Tian, S. F., & Xu, W. (2003, November). Improving one-class SVM for anomaly detection. In *Machine Learning and Cybernetics, 2003 International Conference on* (Vol. 5, pp. 3077-3081). IEEE.

Li, X., Wang, L., & Sung, E. (2008). AdaBoost with SVM-based component classifiers. *Engineering Applications of Artificial Intelligence*, *21*(5), 785–795. doi:10.1016/j.engappai.2007.07.001

Ma, J., & Perkins, S. (2003, July). Time-series novelty detection using one-class support vector machines. In *Neural Networks, 2003.Proceedings of the International Joint Conference on* (*Vol. 3*, pp. 1741-1745). IEEE. doi:10.1109/IJCNN.2003.1223670

Maglaras, L. A., & Jiang, J. (2014). A real time OCSVM Intrusion Detection module with low overhead for SCADA systems. *International Journal of Advanced Research in Artificial Intelligence*, *3*(10).

Maglaras, L. A., & Jiang, J. (2014, August). Intrusion detection in scada systems using machine learning techniques. In *Science and Information Conference (SAI)* (pp. 626-631). IEEE doi:10.1109/SAI.2014.6918252

Maglaras, L. A., & Jiang, J. (2014, August). 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* (pp. 133-134). IEEE. doi:10.1109/QSHINE.2014.6928673

Maglaras, L. A. & Jiang, J. (2015). A novel intrusion detection method based on OCSVM and K-means recursive clustering. *EAI Transactions on Security and Safety*, 1-10.

Maglaras, L. A., Jiang, J., & Cruz, T. (2014). Integrated OCSVM mechanism for intrusion detection in SCADA systems. *Electronics Letters*, *50*(25), 1935–1936. doi:10.1049/el.2014.2897

Maglaras, L. A., Jiang, J., & Cruz, T. J. (2016). Combining ensemble methods and social network metrics for improving accuracy of OCSVM on intrusion detection in SCADA systems. *Journal of Information Security and Applications*.

Menahem, E., Rokach, L., & Elovici, Y. (2013, October). Combining one-class classifiers via meta learning. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management* (pp. 2435-2440). ACM doi:10.1145/2505515.2505619

Mukkamala, S., Janoski, G., & Sung, A. (2002). Intrusion detection using neural networks and support vector machines. In *Neural Networks, 2002. IJCNN'02.Proceedings of the 2002 International Joint Conference on* (*Vol. 2*, pp. 1702-1707). IEEE. doi:10.1109/IJCNN.2002.1007774

Portnoy, L., Eskin, E., & Stolfo, S. (2001). Intrusion detection with unlabeled data using clustering. In *Proceedings of ACM CSS Workshop on Data Mining Applied to Security* (DMSA-2001).

Roesch, M. (1999). Snort: Lightweight Intrusion Detection for Networks. *LISA*, 229-238.

Sadeghi, A. R., Wachsmann, C., & Waidner, M. (2015, June). Security and privacy challenges in industrial internet of things. In *Proceedings of the 52nd Annual Design Automation Conference* (p. 54). ACM. doi:10.1145/2744769.2747942

Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., & Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural Computation*, *13*(7), 1443–1471. doi:10.1162/089976601750264965 PMID:11440593

Shang, W., Li, L., Wan, M., & Zeng, P. (2015, December). Industrial communication intrusion detection algorithm based on improved one-class SVM. In *2015 World Congress on Industrial Control Systems Security (WCICSS)* (pp. 21-25). IEEE. doi:10.1109/WCICSS.2015.7420317

Song, X., Fan, G., & Rao, M. (2008). Svm-based data editing for enhanced one-class classification of remotely sensed imagery. *IEEE Geoscience and Remote Sensing Letters*, *5*(2), 189–193. doi:10.1109/LGRS.2008.916832

Tax, D. M. (2001). One-class classification. TU Delft, Delft University of Technology.

Tsoumakas, G., Katakis, I., & Vlahavas, I. (2004, September). Effective voting of heterogeneous classifiers. In *European Conference on Machine Learning* (pp. 465-476). Springer Berlin Heidelberg.

Unnthorsson, R., Runarsson, T. P., & Jonsson, M. T. (2003, August). Model selection in one-class SVMs using rbf kernels. In *Proc. 16th Int. Congress and Exhibition on Condition Monitoring and Diagnostic Engineering Management*. doi:10.1109/iThings/CPSCom.2011.34

Vasilomanolakis, E., Karuppayah, S., Mühlhäuser, M., & Fischer, M. (2015). Taxonomy and survey of collaborative intrusion detection. *ACM Computing Surveys*, *47*(4), 55. doi:10.1145/2716260

Viinikka, J., Debar, H., Mé, L., & Séguier, R. (2006, March). Time series modeling for IDS alert management. In *Proceedings of the 2006 ACM Symposium on Information, computer and communications security* (pp. 102-113). ACM. doi:10.1145/1128817.1128835

Wang, Y., Wong, J., & Miner, A. (2004, June). Anomaly intrusion detection using one class SVM. In *Information Assurance Workshop, 2004. Proceedings from the Fifth Annual IEEE SMC* (pp. 358-364). IEEE. doi:10.1109/IAW.2004.1437839

Woo, P. S., Kim, B. H., & Hur, D. (2015). Towards Cyber Security Risks Assessment in Electric Utility SCADA Systems. *Journal of Electrical Engineering and Technology*, *10*(3), 888–894. doi:10.5370/JEET.2015.10.3.888

Yang, Y., McLaughlin, K., Sezer, S., Littler, T., Im, E. G., Pranggono, B., & Wang, H. F. (2014). Multi-attribute SCADA-specific intrusion detection system for power networks. *IEEE Transactions on Power Delivery*, *29*(3), 1092–1102. doi:10.1109/TPWRD.2014.2300099

Zhang, R., Zhang, S., Lan, Y., & Jiang, J. (2008). Network anomaly detection using one class support vector machine. In *Proceedings of the International MultiConference of Engineers and Computer Scientists* (Vol. 1).

Zhu, B., Joseph, A., & Sastry, S. (2011, October). A taxonomy of cyber attacks on SCADA systems. In *Internet of things (iThings/CPSCom), 2011 international conference on and 4th international conference on cyber, physical and social computing* (pp. 380-388). IEEE. doi:10.1109/iThings/CPSCom.2011.34

KEY TERMS AND DEFINITIONS

Critical Infrastructure: A term used to describe assets that are essential for the functioning of a society and economy.

IDS (Intrusion Detection System): A device or software application that monitors a network or systems for malicious activity or policy violations.

Kernel Methods: A class of algorithms for pattern analysis, whose best known member is the support vector machine (SVM).

SCADA (Supervisory Control and Data Acquisition): An industrial automation control system at the core of many modern industries.

SVM (Support Vector Machines): Supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis.

SNORT: A free and open source network intrusion prevention system (NIPS) and network intrusion detection system.

Testbed: A platform for conducting rigorous, transparent, and replicable testing of scientific theories, computational tools, and new technologies.