2016 IEEE 40th Annual Computer Software and Applications Conference

A One-Class NIDS for SDN-Based SCADA Systems

Eduardo Germano da Silva*, Anderson Santos da Silva*, Juliano Araujo Wickboldt*,
Paul Smith†, Lisandro Zambenedetti Granville*, Alberto Schaeffer-Filho*

**Institute of Informatics, Federal University of Rio Grande do Sul, Porto Alegre, Brazil*
Email: {eduardo.germano, assilva, jwickboldt, granville, alberto}@inf.ufrgs.br
†Safety and Security Department, Austrian Institute of Technology, Vienna, Austria
Email: paul.smith@ait.ac.at

***Abstract*— Power systems are undergoing an intense process of modernization, and becoming highly dependent on networked systems used to monitor and manage system components. These so-called Smart Grids comprise energy generation, transmission, and distribution subsystems, which are monitored and managed by Supervisory Control and Data Acquisition (SCADA) systems. In this paper, we discuss the benefits of using Software-Defined Networking (SDN) to assist in the deployment of next generation SCADA systems. We also present a specific Network-Based Intrusion Detection System (NIDS) for SDN-based SCADA systems, which uses SDN to capture network information and is responsible for monitoring the communication between power grid components. Our approach relies on SDN to periodically gather statistics from network devices, which are then processed by One-Class Classification (OCC) algorithms. Given that attack traces in SCADA networks are scarce and not publicly disclosed by utility companies, the main advantage of using OCC algorithms is that they do not depend on known attack signatures to detect possible malicious traffic. Our results indicate that OCC algorithms achieve an approximate accuracy of 98% and can be effectively used to detect cyber-attacks targeted against SCADA systems.**

I. Introduction

Electric power grids are undergoing a modernization process and evolving into the so-called Smart Grids [1][2], improving the generation, transmission, and distribution of electrical energy. Smart Grids allow a more resilient, secure, and reliable power supply for end-users, such as industries, schools, hospitals, and residences. A power grid is composed of thousands of electronic devices, such as transformers, relays, fuses, or disconnectors. An important component of a power grid is the Supervisory Control and Data Acquisition (SCADA) system [3], responsible for monitoring, controlling, and managing automated processes of the power grid, such as shutting down of an electrical substation, or monitoring the passing electric tension on a transmission line.

Just as power grids are becoming Smart Grids, SCADA systems are also evolving, for example, by using more secure communication protocols and field devices with more processing capacity. Recently, efforts to merge Software-Defined Networking (SDN) with SCADA systems have been carried out [4][5]. SDN is a promising network paradigm that can support the evolution of SCADA communication networks as well [4]. SDN introduces an architecture that simplifies network operations by relying on a logically centralized element often referred as controller [6]. That adds the flexibility required to quickly deploy and configure new field devices and to develop more complex network services in the context of SCADA networks [4].

Given the importance of power grid infrastructures, they may become target of malware infections and cyber-attacks. If these threats are not properly detected and handled they can cause outages in power supply, or even destroy substation equipment [7]. An Intrusion Detection System (IDS) [8] is then necessary to assist in detecting and mitigating such threats. In this paper, we rely on OpenFlow, currently the most important protocol for SDN implementation [9], to present a Network-based IDS (NIDS) [10] designed specifically for SDN-based SCADA systems. Our NIDS uses One-Class Classification (OCC) algorithms [11] that enable detecting abnormal traffic behavior from a homogeneous training set containing only the signature of traffic generated under normal network operation [11]. There are many aspects that make OCC algorithms ideal for anomaly detection in SCADA networks, which includes: (i) they can detect unknown types of attacks for which there are no signatures available, (ii) they do not require specific SCADA attack traces, which are scarce and often not publicly disclosed by utility companies, and (iii) this class of algorithm is suitable for the kind of traffic behavior and periodicity found in SCADA systems.

To demonstrate the benefits and accuracy of our proposed NIDS, we present an analysis comparing two machine learning algorithms of OCC: OCSVM [12] and SVDD [13]. This comparison shows the efficiency of our approach to detect cyber-attacks targeted at a power grid. In our experiments, we simulated an SDN-based SCADA system using a large-scale topology, with one main control center, four intermediate control centers, eight distribution substations, and hundreds of field devices. SDN enables the SCADA system to control and monitor field devices and the network that interconnects SCADA components. The SCADA system

0730-3157/16 $31.00 © 2016 IEEE
DOI 10.1109/COMPSAC.2016.32

uses Modbus/TCP protocol and, during the experiments, we simulated an attacker that starts a DoS attack targeted at one substation. The cyber-attack exploits a Modbus vulnerability that allows flooding unauthorized read requests to SCADA devices operating under this communication protocol [14].

The remainder of this paper is organized as follows. In Section II, we present background on SDN-based SCADA systems, as well as fundamental concepts related to One-Class Classification algorithms. In Section III, we describe the design of our proposed NIDS and details of the algorithms implemented. In Section IV, we present the evaluation results and a performance analysis of our approach. In Section V, we describe the related work and finally, in Section VI, we conclude this paper with final remarks and future work.

II. Background

In this section, we present the fundamental aspects about SCADA systems and the communication protocols used in this kind of cyber-physical system. In addition, we present how SDN can assist in the modernization process of SCADA systems and power grids. Furthermore, some concepts of OCC are discussed as well as their advantages and benefits in the detection of network anomalies in SCADA systems.

A. SDN-Based SCADA Systems

SCADA systems are used in critical infrastructures such as power grids, water supplies, oil and gas facilities. In power grids, in specific, SCADA systems are highly distributed, used by power utilities to collect data, monitor, and control devices through power lines [3]. SCADA systems present a well-defined architecture with two main types of components: a single master unit, called Master Terminal Unit (MTU); and slave units, or Remote Terminal Units (RTUs) [7]. For large-scale SCADA systems that contain several RTUs, subMTUs are also employed to alleviate the workload on the primary MTU [3]. In the power distribution system, RTUs monitor energy distribution substations and the electrical voltage forwarded to final users [4]. These substations are usually composed of thousands of field devices, such as sensors, circuit breakers, actuators, relays, and transformers [5]. Thus SCADA systems have a large number of interconnected devices that transmit a considerable amount of information about the system actuation environment and the automated process.

To provide a more reliable data transmission mechanism, most of the communication protocols used by current SCADA systems were ported to execute over TCP/IP, *e.g.*, Modbus/TCP and DNP3 over TCP/IP [15]. In these cases, adaptations have been carried out without considering security aspects (*e.g.*, data encryption) allowing an attacker to intercept the communication and read the information being transmitted [4]. Another important issue is that, to allow remote and more flexible maintenance of their components, SCADA systems were typically connected to the corporate network of the organization responsible for the system, that consequently is directly or indirectly connected to the Internet. This tendency makes SCADA systems susceptible to common threats, such as malware and cyber-attacks [15].

There are many efforts to increase the security level of SCADA systems, *e.g.*, proposal of new communication protocols, improvement of existing protocols, new security and management standards, or even IDSes [15][16]. However, the incorporation of SDN into SCADA systems emerges as an attractive research area, since SDN can help in the evolution of SCADA communication networks, facilitating the development of network applications [4]. The adoption of SDN in SCADA will support more resilient systems, as solutions to mitigate attacks and other threats can be more easily implemented in the SDN controller. SCADA systems can benefit form the characteristics of SDN in several ways:

- **Flexibility:** SDN permits easily adding new field devices or upgrading existing network applications inside the SCADA system [5].
- **Centralized Management:** The MTU can not only manage field devices but also monitor and control the network that interconnects system devices [4].
- **Standard API:** The OpenFlow protocol provides a standard API that allows a better integration of geographically disperse network equipment from different vendors [5][17].
- **Programmability:** SDN allows creating a range of customized services, *e.g.*, to perform load balancing between communication links, to optimize the operation of system components, or even to identify and mitigate traffic anomalies [4].

B. One-Class Classification

Anomaly detection techniques aim to detect unexpected behaviors, also called *anomalies* or *outliers*, in a dataset [18]. In communication networks, unexpected behaviors may represent the occurrence of malicious activities, creating a real necessity for sophisticated detection mechanisms to prevent the network from service degradation. Frequently, machine learning is suitable to this task because it offers a wide range of mechanisms that can be applied for traffic classification and for detecting intrusions in different contexts.

There are different classification techniques associated with machine learning. Supervised machine learning algorithms require a training step, *i.e.*, an initial step in which the classifier learns the profile of the target class. A classifier trained with positive and negative classes is called a binary classifier, while a classifier that is trained with several samples representing many classes is named a multiclass classifier. However, both of these types of classifiers require the profile of the target attack to operate, which may be difficult to be obtained when the profile required is a novel, unknown attack. To alleviate this restriction, OCC algorithms have been designed such that they only need one data profile, which in the case of network can be the normal expected traffic behavior, a frequently available traffic information.

SCADA systems are environments that can take advantage of OCC algorithms. It is noteworthy that the same benefits of using machine learning in general networks can be achieved in SCADA systems as well. In addition, SCADA traffic flows are naturally periodic and their networks have stable connection matrices [7]. This characteristic encourages the use of OCC algorithms for detecting anomalous behaviors in SCADA networks. These algorithms do not rely on malicious signatures, but instead need only the expected traffic behavior, making the detection process faster and more accurate. Since an OCC-based NIDS does not require attack signatures to build a classifier model, it is well suited for intrusion detection in SCADA systems [19]. Further, SDN allows the periodic gathering of precise statistics in SCADA networks. These statistics can be used to create a model of the SCADA system normal and expected behavior. Thus, this behavioral model in combination with OCC algorithms, is used to build a resilience mechanism for detecting cyber-threats in SCADA networks.

III. One-Class NIDS for SDN-Based SCADA Systems

In this section, we present a brief background on the OCC algorithms that we adopted in the proposed NIDS for SDN-based SCADA systems. Furthermore, we also introduce our strategy to detect intrusions in SCADA communication networks and detail the architecture and the respective components of our NIDS.

A. OCC Algorithms

The OpenFlow protocol allows the integration of network devices from different vendors through its standard API [4]. This feature facilitates gathering network flow statistics via a logically centralized controller. These statistics about the SCADA network are essential for the proper functioning of our NIDS. Algorithms are used to analyze these statistics, finding anomalous behaviors in the SCADA network. Furthermore, we also consider fundamental that our approach must:

- Adapt to the scale and heterogeneity of existing SCADA systems. In other words, the proposed NIDS must be able to detect anomalous behaviors in SCADA networks of small power distribution companies and large-scale systems responsible for managing the power grid of entire countries, independently of the protocol used or the behavior of network devices;
- Manipulate constantly, promptly, and accurately large datasets. This requirement is important because the sampling period in SCADA systems in the power distribution sector ranges from 2 to 4 seconds [20]. In addition, a single substation may contain thousands of field devices [5] indirectly (via RTU) or directly connected to the SCADA MTU. Furthermore, a NIDS that has a fast anomaly detection process can avoid or minimize the incidence of outages in the power supply;
- Enable the detection of harmful network anomalies in SCADA systems. Our NIDS must detect anomalous behaviors known by operators as well as anomalies that exploit previously unknown vulnerabilities, and consequently unpatched (Zero-Day exploit). Furthermore, as SCADA attack traces are scarce and not publicly disclosed, the proposed NIDS must be able to detect such anomalies without using attack signatures.

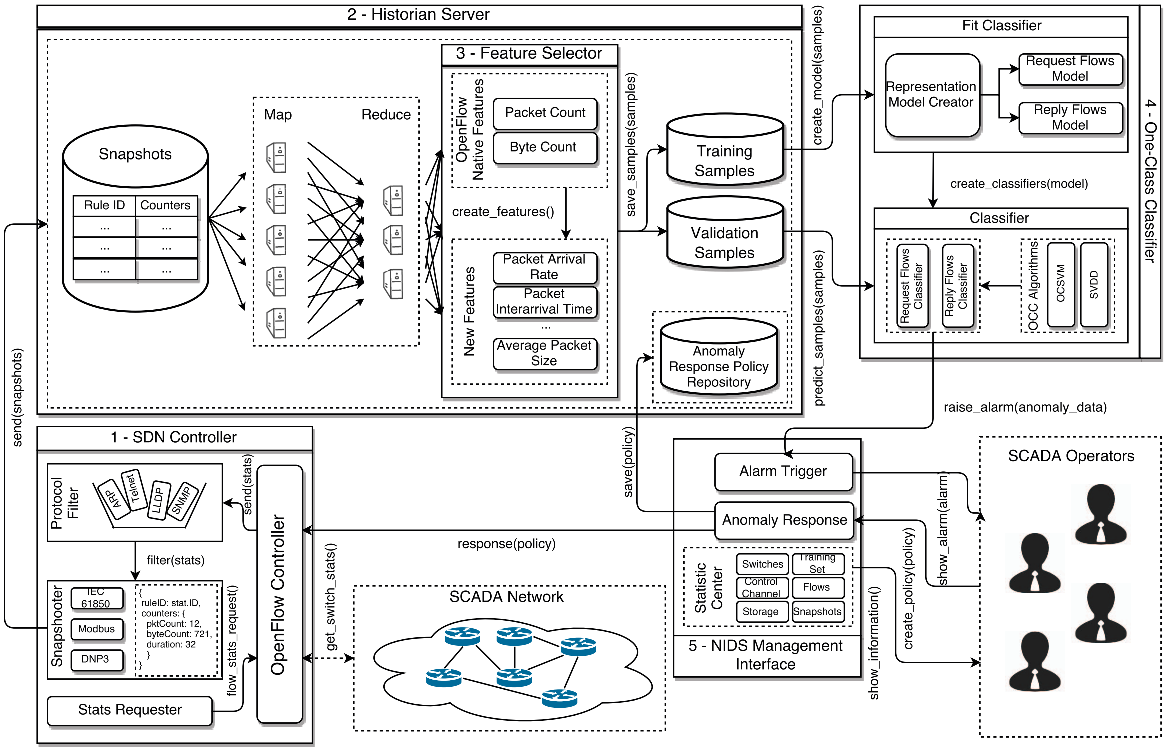
To fulfill these requirements, we adopted OCC algorithms based on Support Vector Machine (SVM), such as One-Class Support Vector Machine (OCSVM) [12] and Support Vector Data Description (SVDD) [13]. SVMs are a set of supervised learning methods that analyze datasets and recognize patterns [21]. SVMs are among the most popular methods of supervised learning. SVM-based algorithms deal well with large datasets and in most cases perform better in comparison to other supervised learning methods [22]. A brief description of OCSVM and SVDD is presented below:

1) One-Class Support Vector Machine (OCSVM): It is a supervised machine learning algorithm presented by Schölkopf *et al.* [12]. OCSVM resembles the traditional two-class SVM, where the training set is composed of two groups, one positive and one negative. However, OCSVM, in its training phase, uses a homogeneous set of instances and is indicated for problems that involve anomaly detection. Thus, with data used to fit the algorithm, OCSVM learns a decision function and classifies new data as similar or distinct to the training set. As the traditional SVM for binary classification, OCSVM uses kernel methods for classifying validation samples. In particular, OCSVM can use traditional kernel methods (linear, polynomial, radial basis function - RBF, or sigmoid), or customized kernels defined by the user.

2) Support Vector Data Description (SVDD): Also known as Support Vector Domain Description, it was introduced by Tax and Duin [13][23] and is another type of SVM-based OCC algorithm. SVDD is a useful method for novelty detection and has been applied to a variety of applications that need to monitor the rise of novelties. This algorithm uses the training set to define a hypersphere with minimum radius, which is used for binary classify samples of a validation set. The hypersphere of SVDD is modeled to involve the majority of training samples. In the validation stage, new samples that are not inside the hypersphere area are classified as novelties, whereas samples that are inside the hypersphere are considered normal samples.

B. Architecture Overview

In this paper, differently from previous approaches that present IDSes for traditional SCADA



```
{
ruleID: stat.ID,
counters: {
  pktCount: 12,
  byteCount: 721,
  duration: 32
  }
}
```

Figure 1. Architecture overview of the proposed NIDS for SDN-Based SCADA Systems.

systems [7][19][24][25], we present a NIDS designed and developed for SDN-based SCADA systems. Our approach takes advantage of the characteristics of SCADA networks to accomplish *novelty* detection in this environment. Considering that SCADA systems have predictable and periodic network traffic and that their networks present a stable connection matrix [7], any novelty can be considered a malicious behavior, such as the beginning of a cyber-attack, or even the misconfiguration of a particular field device. So, as our classifier knows the expected behavior of the SCADA network, any data that indicates novelty in relation to this model can be reported to the operator as a network anomaly. When the entry point of an anomaly in the SCADA network is determined, SDN allows routing rules to be updated in real time. This facilitates anomaly mitigation, enabling the redirection of malicious network traffic, or even dropping the intruder's packets.

The OpenFlow protocol installs rules in each network switch to route data packets. Each rule may have unique information about a communication flow, such as, MAC and IP address of source and destination devices, packet and byte counters, rule duration, and the switch in which the rule is installed. Although there are other approaches for collecting network information, we keep our solution based only on SDN. This allows the proposed NIDS to be extended to create more sophisticated solutions for resilience in SCADA networks. The proposed NIDS uses the OpenFlow protocol for periodically extracting statistics of the SCADA network. This strategy permits continuously gathering statistics on the same frequency of the SCADA sampling period. Our NIDS generates samples from these network statistics. Samples serve as basis to build the model that predicts the network behavior. Each sample has features that are defined by the SCADA operators *a priori*. In addition to the native features provided by OpenFlow (packet count, byte count, and rule duration), our approach also enables the use of other features extracted from native ones. The selection of an optimal set of features can increase classifier accuracy [26], decreasing the rate of false-positive alarms generated during network monitoring.

Our NIDS is composed of five components that inter-communicate to monitor the network and to report possible anomalous behaviors in SCADA systems. Our architecture has been designed to be scalable, *i.e.*, it must allow the detection of anomalies in SCADA systems responsible for monitoring small regions, as well as large-scale systems. Thus, the proposed NIDS must be also capable of promptly processing large amounts of data. For this reason, the architecture contemplates the adoption of mechanisms that enable the parallel and distributed processing of data. Figure 1 presents an overview of the proposed NIDS architecture. A brief description of its components is presented below:

*1) **SDN Controller**:* This component is responsible for monitoring and for applying routing strategies to SCADA network switches. It is composed of *Stats Requester*, *OpenFlow Controller*, *Protocol Filter*, and *Snapshooter*. *Stats*

Requester is a module responsible for creating request messages of flows statistics and for controlling the periodicity in which these messages are sent to network switches. It generates requests in accordance with the periodicity previously defined by the SCADA operator. As said before, OpenFlow is currently the most important protocol for SDN implementation. The *OpenFlow Controller* executes routing strategies, applies anomaly response policies on the network, and manages statistic requests forwarded to the switches and their respective replies. At this stage, *Protocol Filter* blocks statistics regarding non-specific SCADA protocols, allowing only previously defined types of packets (such as LLDP, ARP, and ICMP, and specific SCADA protocols, such as Modbus, DNP3, and IEC-61850) to be forwarded on the network and generate statistics for the NIDS. Finally, *Snapshooter* is responsible for structuring these statistics of SCADA protocols in snapshots and for forwarding this information to the Historian Server.

*2) **Historian Server**:* This is a component that is typically present in the Control Center of several traditional SCADA systems [14]. This server stores logs about SCADA devices. We extended this component to store network snapshots collected by the SDN Controller. It needs to be able to store and process datasets generated by large-scale SCADA systems, and Distributed and parallel processing can help in reducing possible bottlenecks. Thus, in order to allow large-scale data processing, the proposed NIDS relies on the distributed nature of the MapReduce (MR) programming model [27] to implement a scalable version of the Historian Server. MR is used for: (*i*) finding flow rules stored in the Historian Server through the rules header (mapping process); and (*ii*) reducing these headers in keys (reduction process). These keys are fundamental for finding counters of a rule stored in the server. In other words, MR is used to find active rules in the network and catalog their counters. The stored *snapshots* are converted into *Training Samples* (used to fit the classifier model), and *Validation Samples* (analyzed by the NIDS to monitor the SCADA network). To increase its classification accuracy, we split samples in two classes: request and reply samples. Request samples have the MTU as source device, whilst reply samples have an RTU as source device. Further, the Historian Server also has the *Anomaly Response Policy Repository*, which is a component that contains strategies for anomaly mitigation predefined by SCADA operators. Thus, our NIDS can act proactively on the network without direct intervention of an operator, for example: redirecting an anomalous flow to a HoneyPot device; reducing the priority of a harmful flow to the system; or dropping malicious network packets.

*3) **Feature Selector**:* OpenFlow only provides native flow features, namely packet count, byte count, and rule duration. These features may not be adequate to describe the nature of a specific traffic profile. Using appropriate features to describe traffic behavior may increase the accuracy of our NIDS. Thus, the *Feature Selector* component [26] analyses the stored samples and offers an extensive set of features extracted from OpenFlow native counters, such as packet inter-arrival-time, packets per second, mean packet length, and so forth. Besides, this component uses feature selection techniques, such as Principal Component Analysis (PCA) and Genetic Algorithm (GA), to determine the optimal set of features for traffic classification. We strategically placed this component inside the Historian Server in order for it to have easy access to the stored traffic samples. Note that, however, we do not incorporate this component in our prototype and preliminary experiments, and this task is proposed as future work.

*4) **One-Class Classifier**:* This is the central component in the proposed architecture as it analyzes samples to find anomalous behaviors in the SCADA network. At first, *Fit Classifier* is responsible for the training step of the proposed NIDS. This training step is defined by the SCADA operator and occurs before the validation stage. The SCADA operator also can define if the training stage will occur from trace files, or from the normal functioning of the network. In addition, it is possible to define the periodicity in which our NIDS will be trained again. This component receives training samples for generating classification models: one specific for request flows, and another specialized in reply flows. These classification models are forwarded to the *Classifier* component, which in turn generates two classifiers: Request Classifier and Reply Classifier. Our approach allows SCADA operators to choose and combine OCC algorithms available for traffic classification. In addition, if any validation sample indicates an inconsistent state in the SCADA system, the Classifier will send information about the unexpected behavior to the NIDS Management Interface.

*5) **NIDS Management Interface**:* This component provides a management interface for SCADA operators to interface with our NIDS. NIDS Management Interface receives the details of anomalies detected by the One-Class Classifier and generates alarms to the operators. Alarms generated by the *Alarm Trigger* component contain information about where and when the network anomaly occurred in the power system. The *Anomaly Response* component allows the operator to define response policies to the anomalous behavior detected. These policies are stored in the Historian Server, in Anomaly Response Policy Repository, to be reused after. In addition, these policies are directly forwarded to the SDN Controller that applies these actions over the SCADA network. Management policies provided by SCADA operators can be used to redirect a DDoS attack to a HoneyPot device, and block or limit the anomalous traffic through an OpenFlow rule. Finally, *Statistic Center* presents information about the SCADA network and the Historian Server, such as the traffic on the OpenFlow control channel, communication flows disposed in the network, amount of snapshots stored in the database, and even the size of the training set.

IV. Prototype and Experimental Results

In this section, we describe the proof-of-concept prototype that we implemented, the experimental setup used for the evaluation of the OCC algorithms, as well as the test scenarios used to simulate a realistic SCADA environment. We also discuss the experimental results that validate our prototype and verify the accuracy of the classification techniques.

A. Prototype and Experimental Setup

We chose the POX SDN/OpenFlow Controller[1], which uses OpenFlow version 1.0 to manage and monitor the SCADA communication network. We implemented the Historian Server on a NoSQL database, more specifically the Apache Cassandra Server[2]. This database promotes design scalability and allows distributing system tasks to multiple clusters, decreasing possible processing bottlenecks. Finally, we used the LIBSVM[3] library that offers a simple and efficient implementation of the necessary machine learning algorithms, OCSVM and SVDD, for our prototype. It is important to note that both algorithms were used with their default parameters. Moreover, we used the RBF Kernel for the OCSVM algorithm.

We defined an evaluation scenario that simulates the SDN-based SCADA system of a particular power grid company. This company controls a hydro-power plant, transmission lines, and eight distribution substations. This power grid is responsible for supplying a particular region. To control and monitor this power grid, the company has a SCADA system based on a large-scale topology, with one MTU, four subMTUs, eight RTUs (each RTU contains 750 field devices directly connected), and eight hundred independent field devices. The independent field devices were simulated to control the voltage in transmission lines and each one was simulated independently. In addition, each independent field device and each RTU were subordinated to one subMTU. The MTU requests data from its subMTUs, that consequently request data from their subordinated devices (RTUs or field devices). We used the TCP version of the most common SCADA protocol, Modbus [28], for communicating devices of our SCADA system. To do that, we used the PyModbus library[4] and its Modbus `READ COILS` function. This function allows a MTU to read values of RTUs registers. Figure 2 depicts the topology that we used in our evaluation scenario. To simulate this environment, we used Mininet[5]. The topology, number of substations, number of field devices and protocols used are based on documents reporting actual SCADA systems deployed in the US [3].

Our experiments consisted of the MTU and subMTUs periodically requesting information from the subordinated devices, every 2 seconds. To respect the time requirements of traditional SCADA systems for power grids, we set the periodicity of network statistics gathering to 4 seconds. Furthermore, our NIDS was configured to collect information from the Historian Server every 4 seconds too. Further, the NIDS was responsible for analyzing the network information stored in the Historian Server and for generating alerts to system operators when a possible anomalous behavior in the SCADA communication network was detected.

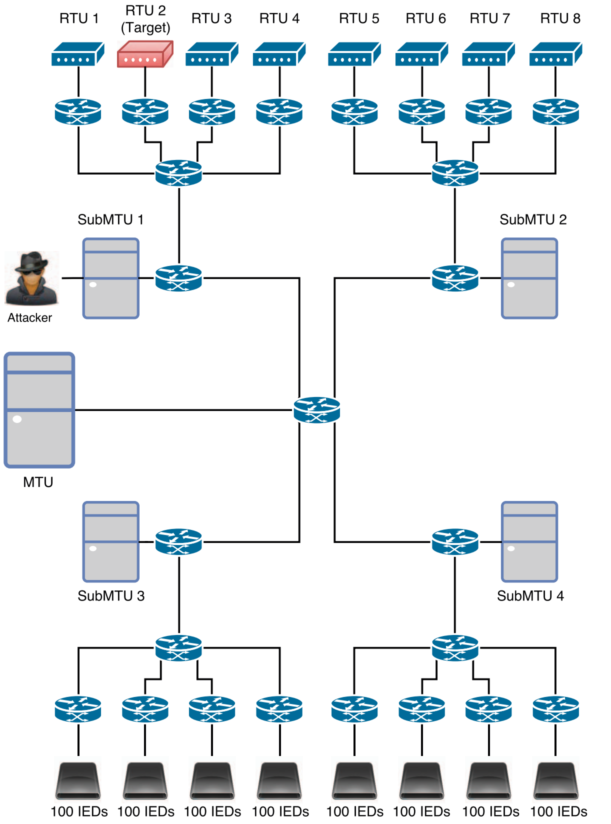


Figure 2. Configuration of the network topology used in our experiments.

During the experiments, we simulated the launch of a DoS attack targeted at one of the substations. We assume that this DoS could be launched by a disgruntled employee of the power distribution company who has remote access to one workstation of subMTU #1. The attacker intends to disrupt the communication between a particular substation (RTU #2) and the rest of the power system to force a hard-reset in the SCADA system and to cause financial losses to the company. The DoS attack is based on a vulnerability of Modbus/TCP named Unauthorized Read Request[6]. This vulnerability allows an attacker with IP connectivity to the RTU to send unlimited data requests and consequently cause a buffer-overflow in the SCADA slave device [29]. Currently, buffer-overflow is a major threat in modern and legacy SCADA systems [30]. We collected 24 hours of training

[1] http://www.noxrepo.org/pox

[2] http://cassandra.apache.org/

[3] https://www.csie.ntu.edu.tw/∼cjlin/libsvm/

[4] https://code.google.com/p/pymodbus/

[5] http://www.mininet.org/

[6] http://www.symantec.com/security_response/attacksignatures

samples to build a classifier model with the daily behavior of the evaluation scenario. Each experiment was conducted during 20 minutes and they contained two types of traffic: 10 minutes represented the expected system functioning; and 10 bursts of 1 minute indicated the occurrence of the DoS attack on the SCADA network.

B. Evaluation Results

We evaluated the quality of the proposed NIDS for SDN-based SCADA systems using the two available SVM-based OCC algorithms for traffic classification, OCSVM and SVDD. With the assistance of a confusion matrix, calculated from the results of our experiments, we analyzed specific metrics for supervised machine learning algorithms, such as true positive rate (TPR), true negative rate (TNR), positive predictive value (PPV), negative predictive value (NPV), false positive rate (FPR), false discovery rate (FDR), false negative rate (FNR), and accuracy (ACC) [31]. In absolute numbers, our experiments generated a total of 29,709 samples which were classified by the proposed NIDS. Of the total of samples generated, 15,579 (52.439%) represented the normal functioning of our SCADA network, whilst the 14,130 (47.561%) remaining samples evidenced the DoS propagation in the power grid. Each repetition of our experiment generated on average 990.3 validation samples, which results in 519.3 positive samples and 471 negative samples on average for each repetition. It is important to note that the experiments were performed 30 times, in order to achieve a confidence level of 95%.

The confusion matrices presented in Figure 3 describe the traffic classification produced by the proposed NIDS. With both algorithms, OCSVM and SVDD, we can observe that our NIDS obtained significant results if we consider the validation samples classified as false-positive (FP) and true-negative (TN). The confusion matrices show that, for all repetitions of the experiments, our NIDS detected correctly and instantly the validation samples that indicated the propagation of the DoS attack on the SCADA network. In other words, 100% of the validation samples that presented the evidence of DoS attack (TN bars in Figure 3) were classified as anomalous behavior, being reported immediately to the SCADA operators. However, if we only analyze the expected behavior, we can see that OCSVM obtained slightly better performance if compared to the SVDD algorithm. OCSVM classified correctly, that is, classified as true-positive (TP), approximately 98.435% of the validation samples, whilst using SVDD this classification was 95.718%. Figure 4 presents a time series of the traffic classification of both OCSVM (Figure 4(a)) and SVDD (Figure 4(b)), respectively. As Figure 4(b) indicates, in the experiment using SVDD, the last three validation samples were classified as FP. This indicates that the RBF Kernel used by OCSVM fits better to the simulated traffic in our SCADA system in relation to the SVDD hypersphere. Note that figures 4(a) and 4(b), for better visualization of the traffic classification, present only part of an experiment, which contains 40 validation samples where 24 samples are the normal expected traffic, and 16 are the traffic samples evidencing the DoS attack.

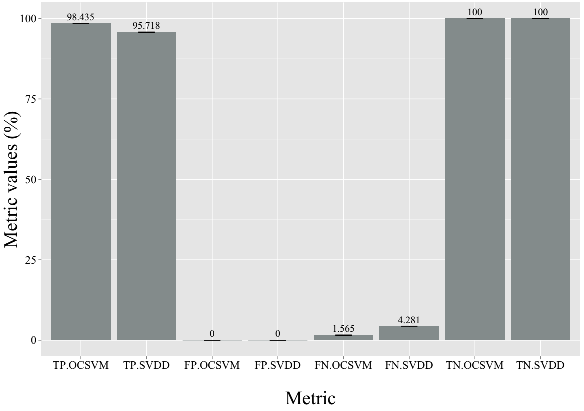


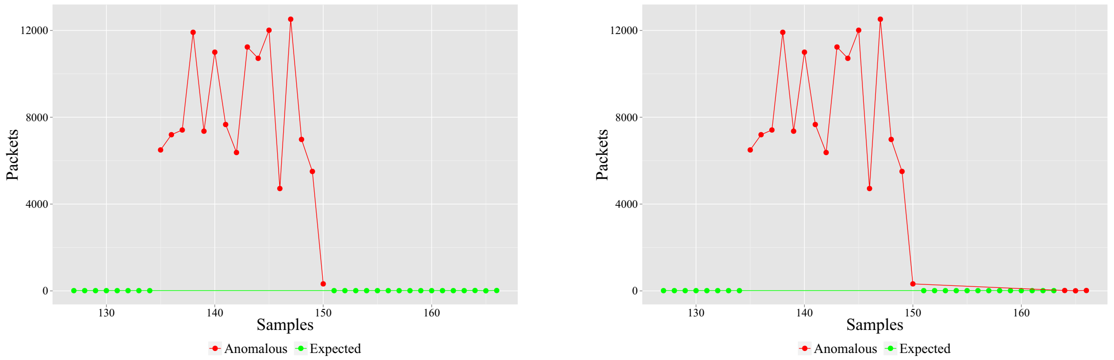
Figure 3. Confusion matrices generated through of the experiments.

As both algorithms obtained similar FP and TN rates, these results also have influence on metrics for evaluating the accuracy of our NIDS. As can be seen in Figures 5(a) and 5(b), both algorithms achieved 100% of TNR and PPV. Consequently, OCSVM and SVDD also achieved 0% of FPR and FDR. These results show that the One-Class NIDS accurately detected and reported the DoS attack in our evaluation scenario. Analyzing the remaining metrics, we can see that SVDD presented a higher FNR (4.281%) if compared to OCSVM (1.565%). OCSVM presented slightly higher TPR (98.435% against 95.718%). Furthermore, OCSVM also presented higher rate of NPV (98.31% against 95.501%). Ultimately, OCSVM obtained slightly better accuracy (99.18%) than SVDD (97.755%). Table I presents an overview of the obtained results in our experiments.

Table I
Overview of results.

Metrics / Algorithm	**OCSVM**	**SVDD**
TPR	$\approx 98.435\%$	$\approx 95.718\%$
TNR	100%	100%
PPV	100%	100%
NPV	$\approx 98.31\%$	$\approx 95.501\%$
FPR	0%	0%
FDR	0%	0%
FNR	$\approx 1.565\%$	$\approx 4.281\%$
ACC	$\approx 99.18\%$	$\approx 97.755\%$

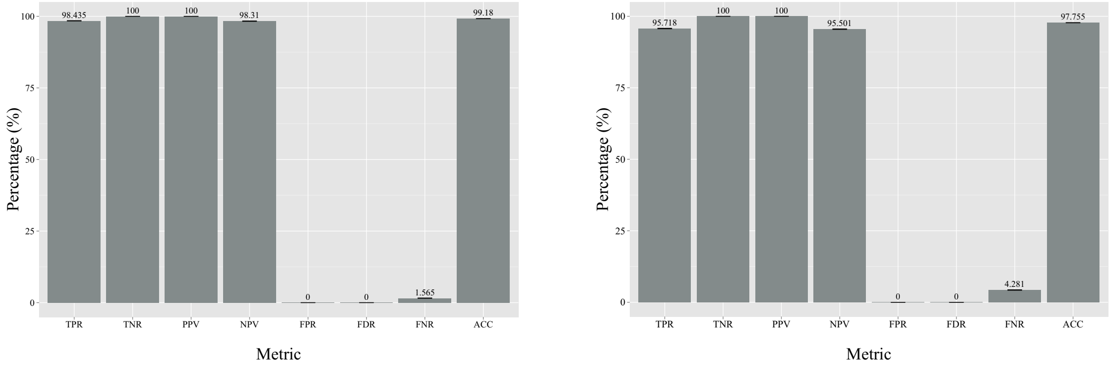
Finally, analyzing the final results, we can state that: novelty detection can be used in the detection of network anomalies in SCADA systems; and in our evaluation scenario, our approach was able to fully detect the DoS attack. Although the proposed NIDS has classified validation samples as FP, it behaved well in relation to the attack detection. This can be proved by analyzing the results obtained on metrics directly based on TN samples or FP samples, such as TNR, PPV, FPR, and FDR. Our experiments also showed



(a) Traffic classification using OCSVM.

(b) Traffic classification using SVDD.

Figure 4. Traffic classification of our One-Class NIDS for SDN-Based SCADA systems.



(a) Metrics using OCSVM.

(b) Metrics using SVDD.

Figure 5. Machine learning metrics obtained from our experiments.

that OCSVM presented slightly better accuracy than SVDD. This means that, given our assumption about the scarcity of public available attack traces in SCADA, it is possible to detect attacks targeted at the power system by using a classifier model that only represents the expected behavior of the SCADA network.

V. Related Work

In this section, we review research efforts that are related to our work. In Section V-A, we describe research efforts that employ SDN in Smart Grids and SCADA systems. In Section V-B, we present studies that aim to detect possible intrusions in SCADA systems. Finally, in Section V-C, we discuss our contribution to this research topic.

A. *SDN in Smart Grids and SCADA systems*

Research efforts investigating the use of SDN in SCADA systems are still scarce. In a previous work, we discussed the potential benefits that SDN can bring to the electrical grid and SCADA systems [4]. The aforementioned paper presents a multipath approach for SDN-based SCADA systems in which communication of SCADA devices is performed by more than one route in order to prevent possible eavesdroppers from fully capturing messages exchanged between SCADA devices. Cahn *et al.* [5] discussed how SDN can alleviate some of the current problems in Smart Grid communication networks. The authors presented the design and development of a new architecture for communication with grid substations, allowing the network to be auto-configurable, secure, and reliable against possible system misconfigurations. Kim *et al.* [32] introduced an SDN-based architecture that simplifies the development of network applications for Smart Grids, enabling self-configuration, security, and scalability. Dorsch *et al.* [33] presented and analyzed advantages and challenges of applying SDN in distribution and transmission networks of Smart Grids. Moreover, that work also introduced algorithms for fast recovery and load management, tested in IEC-61850 traffic. Gyllstrom *et al.* [34] also developed and evaluated algorithms for SDN-based Smart Grid networks. That paper analyzed the performance of algorithms for fast recovery facing link failures. Finally, Goodney *et al.* [17] proposed the use of SDN to control the communication between devices responsible for measuring electrical waves in the grid, known as Phasor Measurement Units (PMUs).

B. *IDS for SCADA Systems*

In the literature, there are several research efforts that proposed IDSes for SCADA systems. Almalawi *et al.* [25]

proposed an unsupervised SCADA data-driven anomaly detection approach intended to be used as a passive SCADA IDS. This IDS has two main steps: (*i*) the identification of consistent and inconsistent states from unlabeled SCADA data traffic generated by system sensors and actuators using the density factor for the k-nearest neighbors of the observation; and (*ii*) the extraction of proximity-based detection rules for normal and anomalous behavior using statistically determined micro-clusters. Barbosa [7] investigated the main traffic characteristics in SCADA networks and presented a NIDS capable of detecting data injection and DoS attacks. This NIDS specifically explores the periodicity of traffic generated in SCADA systems. Finally, Cheung *et al.* [24] used the network behavior of SCADA components to create behavioral models for traffic analysis and anomaly detection.

C. Discussion

In this paper, we advocate that the use of SDN will enhance the scalability and improve the flexibility of SCADA systems, and will also facilitate the creation of SCADA resilience mechanisms. We investigated IDSes currently available for SCADA systems and the network behavior of SCADA devices to contribute to this research topic. Unfortunately, there are few IDSes that exploit the network behavior of SCADA, and the systems that use these aspects do not employ the characteristics of SDN to collect more accurate information of the network. We proposed a NIDS that creates signatures of the expected network functioning, *e.g.*, it generates behavior models of the correct functioning of SCADA devices. Our approach periodically verifies the SCADA network, through SDN, in order to find anomalous behaviors that differ from the expected SCADA behavior. This verification is made by OCC-based machine learning algorithms. This choice was based on the characteristics of OCC algorithms, on the requirements of periodicity and connection matrix stability of SCADA systems, and on the paucity of SCADA attack signatures publicly available to research.

VI. Conclusion and Future Work

Modern power systems comprise energy generation, transmission, and distribution subsystems that are monitored and managed by large-scale SCADA systems. Because of its importance, any threat to SCADA system operation may result in heavy economical losses or even put lives in danger. Therefore, in order to promote the modernization process of power grids, we investigate the development of a new generation of SCADA systems, named SDN-based SCADA systems. By relying on the global view of SDN-based SCADA systems and on their ability to gather switch statistics, we presented a specific NIDS for this kind of environment. Our NIDS uses OCC machine learning algorithms that, with a unique inlier homogeneous training set, can detect anomalous behaviors in SCADA networks, such as unauthorized system or network activity. We presented experimental results in a realistic SCADA environment that validate our prototype and verify the accuracy of the classification techniques, applied to the detection of a DoS attack based on a real vulnerability of the Modbus/TCP protocol. Our analysis was based on a comparison of two OCC algorithms, OCSVM and SVDD, which deal well with large datasets and have a fast classification process if compared to other machine learning techniques.

As future work, we intend to carry out a performance analysis of our NIDS using other OCC algorithms, such as, Kernel Principal Component Analysis (KPCA) [35] and One-Class Random Forests (OCRF) [36]. We also intend to combine classifiers to improve the accuracy of our proposal. In addition, we intend to incorporate the Feature Selector component in our solution to minimize possible false alerts. Finally, we plan to implement a user-friendly interface, for defining alternatives to mitigate the anomalous behaviors without compromising the functioning of SCADA devices.

Acknowledgement

This work is supported by ProSeG - Information Security, Protection and Resilience in Smart Grids, a research project funded by MCTI/CNPq/CT-ENERG # 33/2013.

References

[1] W. Wang, Y. Xu, and M. Khanna, "A Survey on the Communication Architectures in Smart Grid," *Computer Networks*, vol. 55, no. 15, pp. 3604–3629, Oct. 2011.

[2] Y. Yan, Y. Qian, H. Sharif, and D. Tipper, "A Survey on Smart Grid Communication Infrastructures: Motivations, requirements and challenges," *Communications Surveys Tutorials, IEEE*, vol. 15, no. 1, pp. 5–20, First 2013.

[3] K. Stouffer, J. Falco, and K. Scarfone, "Guide to Industrial Control Systems (ICS) Security," *NIST special publication*, pp. 800–82, 2011.

[4] E. Silva, L. Knob, J. Wickboldt, L. Gaspary, L. Granville, and A. Schaeffer-Filho, "Capitalizing on SDN-Based SCADA Systems: An anti-eavesdropping case-study," in *Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on*, May 2015, pp. 165–173.

[5] A. Cahn, J. Hoyos, M. Hulse, and E. Keller, "Software-Defined Energy Communication Networks: From substation automation to future smart grids," in *Smart Grid Communications (SmartGridComm), 2013 IEEE International Conference on*, Oct 2013, pp. 558–563.

[6] J. Wickboldt, W. Jesus, P. Isolani, C. Both, J. Rochol, and L. Granville, "Software-Defined Networking: Management requirements and challenges," *Communications Magazine, IEEE*, vol. 53, no. 1, pp. 278–285, January 2015.

[7] R. Barbosa, "Anomaly Detection in SCADA Systems: A network based approach," Ph.D. dissertation, University of Twente, Enschede, April 2014. [Online]. Available: http://doc.utwente.nl/90271/

[8] H. Liao, C. Lin, Y. Lin, and K. Tung, "Intrusion Detection System: A comprehensive review," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 16 – 24, 2013.

[9] N. Feamster, J. Rexford, and E. Zegura, "The Road to SDN," *Queue*, vol. 11, no. 12, pp. 20:20–20:40, Dec. 2013.

[10] J. Davis and A. Clark, "Data Preprocessing for Anomaly based Network Intrusion Detection: A review," *Computers & Security*, vol. 30, no. 67, pp. 353 – 375, 2011.

[11] S. Khan and M. Madden, "A Survey of Recent Trends in One Class Classification," in *Artificial Intelligence and Cognitive Science*, ser. Lecture Notes in Computer Science, L. Coyle and J. Freyne, Eds. Springer Berlin Heidelberg, 2010, vol. 6206, pp. 188–197.

[12] B. Schölkopf and A. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2001.

[13] D. Tax and R. Duin, "Support Vector Data Description," *Machine Learning*, vol. 54, no. 1, pp. 45–66, 2004.

[14] B. Zhu, A. Joseph, and S. Sastry, "A Taxonomy of Cyber Attacks on SCADA Systems," in *Proceedings of the 2011 International Conference on Internet of Things and 4th International Conference on Cyber, Physical and Social Computing*, ser. ITHINGSCPSCOM '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 380–388.

[15] V. Igure, S. Laughter, and R. Williams, "Security Issues in SCADA Networks," *Computers & Security*, vol. 25, no. 7, pp. 498 – 506, 2006.

[16] L. Jing, Y. Xiao, S. Li, W. Liang, and C. Chen, "Cyber Security and Privacy Issues in Smart Grids," *Communications Surveys Tutorials, IEEE*, vol. 14, no. 4, pp. 981–997, Fourth 2012.

[17] A. Goodney, S. Kumar, A. Ravi, and Y. Cho, "Efficient PMU Networking with Software Defined Networks," in *Smart Grid Communications (SmartGridComm), 2013 IEEE International Conference on*, Oct 2013, pp. 378–383.

[18] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly Detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 15:1–15:58, Jul. 2009.

[19] P. Nader, P. Honeine, and P. Beauseroy, "Intrusion Detection in SCADA Systems using One-Class Classification," in *Signal Processing Conference (EUSIPCO), 2013 Proceedings of the 21st European*, Sept 2013, pp. 1–5.

[20] M. Hadley and K. Huston, "Secure SCADA Communication Protocol Performance Test Results," *Pacific Northwest National Laboratory (August 2007)*, 2007.

[21] C. Cortes and V. Vapnik, "Support-Vector Networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[22] R. Caruana and A. Niculescu-Mizil, "An Empirical Comparison of Supervised Learning Algorithms," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06. New York, NY, USA: ACM, 2006, pp. 161–168.

[23] D. Tax and R. Duin, "Support Vector Domain Description," *Pattern Recognition Letters*, vol. 20, no. 1113, pp. 1191 – 1199, 1999.

[24] 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, 2007, pp. 1–12.

[25] A. Almalawi, X. Yu, Z. Tari, A. Fahad, and I. Khalil, "An Unsupervised Anomaly-Based Detection Approach for Integrity Attacks on SCADA Systems," *Computers & Security*, vol. 46, no. 0, pp. 94 – 110, 2014.

[26] A. Silva, C. Machado, R. Bisol, L. Granville, and A. Schaeffer-Filho, "Identification and Selection of Flow Features for Accurate Traffic Classification in SDN," in *Network Computing and Applications (NCA), 2015 IEEE 14th International Symposium on*, Sept 2015, pp. 134–141.

[27] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, Jan. 2008.

[28] P. Huitsing, R. Chandia, M. Papa, and S. Shenoi, "Attack Taxonomies for the Modbus Protocols," *International Journal of Critical Infrastructure Protection*, vol. 1, pp. 37 – 44, 2008.

[29] A. Wermann, M. Bortolozzo, E. Silva, A. Schaeffer-Filho, L. Gaspary, and A. Barcellos, "ASTORIA: A framework for attack simulation and evaluation in smart grids." in *Network Operations and Management Symposium (NOMS), 2016 IFIP/IEEE*, April 2016, to appear.

[30] D. Incorporated, "Dell Security Annual Threat Report," Dell Incorporated, Tech. Rep., 2015. [Online]. Available: https://software.dell.com/whitepaper/dell-network-security-threat-report-2014874708

[31] D. M. Powers, "Evaluation: From precision, recall and f-measure to roc, informedness, markedness and correlation," 2011.

[32] Y. Kim, K. He, M. Thottan, and J. Deshpande, "Virtualized and Self-Configurable Utility Communications Enabled by Software-Defined Networks," in *Smart Grid Communications (SmartGridComm), 2014 IEEE International Conference on*, Nov 2014, pp. 416–421.

[33] N. Dorsch, F. Kurtz, H. Georg, C. Hagerling, and C. Wietfeld, "Software-Defined Networking for Smart Grid Communications: Applications, challenges and advantages," in *Smart Grid Communications (SmartGridComm), 2014 IEEE International Conference on*, Nov 2014, pp. 422–427.

[34] D. Gyllstrom, N. Braga, and J. Kurose, "Recovery from Link Failures in a Smart Grid Communication Network using OpenFlow," in *Smart Grid Communications (SmartGridComm), 2014 IEEE International Conference on*, Nov 2014, pp. 254–259.

[35] H. Hoffmann, "Kernel PCA for Novelty Detection," *Pattern Recognition*, vol. 40, no. 3, pp. 863 – 874, 2007.

[36] C. Désir, S. Bernard, C. Petitjean, and L. Heutte, "One Class Random Forests," *Pattern Recognition*, vol. 46, no. 12, pp. 3490 – 3506, 2013.