2015 IEEE Symposium Series on Computational Intelligence

Potentials of Using One-class SVM for Detecting Protocol-specific Anomalies in Industrial Networks

Franka Schuster, Andreas Paul, René Rietz, Hartmut König
Brandenburg University of Technology
Cottbus-Senftenberg, Germany
Email: {franka.schuster,andreas.paul,rene.rietz,hartmut.koenig}@b-tu.de

***Abstract*—Support vector machines (SVM) have been considered for real-life machine learning applications in various fields. Security concerns in modern industrial networks, also used in critical infrastructures, require novel monitoring techniques applicable for these constrained, real-time environments. Characteristics of these networks' traffic indicate that SVM can be a powerful tool for realizing a self-configuring monitoring for industrial infrastructures regarding attacks as kind of anomalies. This paper presents the experimental results of applying one-class SVM (OCSVM) on a number of real-world industrial traffic traces from very different industrial control systems (ICS). Initially focusing on a few network packet attributes, the results are discussed in terms of f-score, precision, and recall for different mappings of the features. The results demonstrate the high potential of using one-class SVM for monitoring packets and packet sequences in these networks.**

I. Motivation

Industrial networks have undergone some crucial trends that lead to a rising exposition of these networks to common IT vulnerabilities and indirect connections to public networks. At the same time, open standards on control and field level have been introduced to allow interoperability between industrial devices among vendors and system layers. Industrial Ethernet (IE), for instance, is meanwhile also widely used in industrial networks, although it lacks essential security features, such as authentication and encryption.

Security measures in this field did not keep pace with these trends. Usually, only firewalls are deployed to protect the industrial networks from external threats. A security monitoring within these networks, regarding internal misuse or attacks that have already circumvented the firewall protection, is rare. Recent incidents [1] prove both the vulnerability of industrial installations as well as the presence of parties eager to seriously harm these infrastructures. In case of a critical infrastructure, this may affect even larger parts of society.

Therefore, in crucial environments access control by firewalls should be complemented by a continuous monitoring within the industrial network segments. The configuration of a convenient monitoring system for these constrained networks is a challenging task. So far, this issue prevents the integration of monitoring techniques in productive environments. We are investigating suitable methods for implementing a self-learning, i.e., self-configuring anomaly detection for industrial network data. Industrial networks are predestined for machine learning applications due to the homogeneity of industrial traffic compared to standard IT networks [2], which is a key issue for the learnability of network data.

We identify one-class support vector machines (OCSVM) [3] as the most promising machine learning concept for this task. For this reason, we investigate in the application of OCSVM in order to implement a feasible self-configuring anomaly detection. The main contributions of this paper are: (1) We measure the quality of OCSVM models that can be constructed from industrial traffic traces by use of only a few packet attributes to show the learnability of industrial traffic for a self-configuring anomaly detection using OCSVM. (2) We compare the detection results using different mappings of packet sequences to OCSVM feature vectors for identifying the most promising modeling for practical use.

The remainder of the paper is organized as follows: In Section II we reason the choice of OCSVM for anomaly detection in industrial networks and emphasize the main questions towards an efficient mapping of industrial traffic to OCSVM feature vectors. After a presentation of the datasets used for the experiments in Section III, we explain in Section IV the different types of mappings applied for training and validation of several OCSVM models as well as the evaluation criteria used. These criteria are measured in a series of experiments, whose results are presented in Section V. After putting our work in the context of other research in Section VI, we conclude with an outlook on future research in this field.

II. Approach and Experimental Questions

In this section, we reason the choice of OCSVM as machine learning concept and introduce our approach for an anomaly detection in industrial networks. We then work out associated questions that shall be answered by experiments.

A. OCSVM for Network Anomaly Detection

The concept of OCSVM is an important machine learning approach for one-class classification. We have chosen it for practical and theoretical concerns.

1) Practical Concerns: One-class classification is used for two-class problems in which only one of the two classes can be described well. The aim is to distinguish between a set of training objects, in our case instances of normal traffic, and all other possible instances as *outliers*, in our case anomalous traffic samples. The resulting *outlier detection* (or *novelty*

978-1-4799-7560-0/15 $31.00 © 2015 IEEE
DOI 10.1109/SSCI.2015.22

detection) is in our context used for *anomaly detection*. From a practical perspective we look for a one-class classification method, since we aim to develop an intrusion detection method for which no sample of attacks or other anomalous traffic is needed. This characteristic is crucial for a feasible intrusion detection in industrial networks, where both almost no attack data is available and experiments with artificial malicious traffic in these constrained installations are unimaginable.

2) Theoretical Concerns: One-class classification methods can be categorized according to the use of one of the following principles: density estimation, reconstruction, or direct boundary estimation.

Density Methods. These methods depend on an estimation of the target class to be learned, i.e., in our case instances describing normal network traffic. This requires both a density estimate for the complete feature space and the training set to be a typical sample from the true data distribution. Our approach shall work well without extensive knowledge about the normal traffic's characteristics and especially in absence of examples of concrete attacks or other anomalous traffic. Consequently, the application of density methods, such as the Gaussian model [4] or a Parzen density estimator [5], is not suitable for our approach.

Reconstruction Methods. They rely on the assumption that outlier objects do not satisfy the distribution characteristics of the training objects. The compression methods applied to a test object involve a reconstruction error. This error is used as a distance to the trained class and test objects with high reconstruction error are classified as outliers. Reconstruction methods usually depend on a comparatively high number of user-defined parameters that are not intuitive beforehand. For instance, a poor choice of the number of hidden units and learning rates in auto-encoder [6] and diabolo networks [7] can lead to a large bias for a certain problem causing the methods to become useless. Consequently, we initially look for a method with less configuration effort. Additionally, another important criterion that many reconstruction methods do not fulfill is the robustness against outliers in the training data. Among these methods are the principal component analysis (PCA) [4], k-means clustering [4], self-organizing maps (SOM) [8], and learning vector quantization (LVQ) [9].

Boundary Methods. The aforementioned methods model various characteristics of the training data in order to derive distance or resemblance measures for outlier identification. For our purpose, however, we just need a clear distinction between normal and anomalous traffic samples without the need (and effort) for accurately modeling the normal traffic itself. Fortunately, there are methods that can directly calculate a boundary around the training set, i.e., the normal traffic samples. Among these boundary methods are the k-centers method [10], the data description based on nearest neighbor distances (NN-d) [11], the support vector data description (SVDD) [11], and one-class support vector machines (OCSVM, ν-SVM) [12]. Whereas SVDD finds a boundary around the training set, OCSVM determines a hyperplane that separates the training set from the origin with maximal margin. Both approaches give similar results. If the data is preprocessed to have unit norm, they are even identical [11]. For our approach, we need a method that is robust against outliers in the training set since we cannot completely preclude the existence of anomalous packets and packet sequences in the learning traffic captured from an industrial site. OCSVM and SVDD are the most promising methods, because they show this robustness more than other boundary methods, such as the k-centers method and the NN-d [11]. Due to the similarities of OCSVM and SVDD results, we decided for one of the methods. We chose OCSVM because powerful implementations are available [13].

B. Our Approach

The idea of using OCSVM for learning normal traffic of an industrial network is depicted in Figure 1. Listening on a network interface, such as the mirror port of a switch, relevant features of each packet k are extracted by a so-called *deep packet inspection* (DPI) module and converted to a numerical representation, i.e., the feature vector of packet k. For modeling sequences of packets, this single-packet vector is used to form a larger vector describing the features and the sequence of multiple packets. These n-packets vectors, in case of monitoring sequences of n packets, can then be used as feature vectors for the construction of an OCSVM model describing normal traffic of the monitored industrial network segment. After learning normal traffic in the training phase, the trained OCSVM model is then used in the anomaly detection phase for the decision whether packet sequences seen are normal or anomalous. Since we aim at a protocol-specific monitoring, our approach requires an analysis of the intended industrial protocols regarding packet structure and common packet sequences. However, the more effort is put into this aspect of the learning, the less tuning of the learning is later required when the detector shall be deployed in real networks.

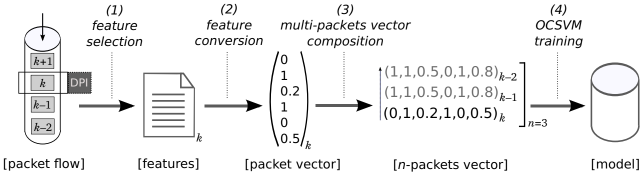

Fig. 1: Schematic depiction of the learning approach

C. Experimental Questions

As with all learning-based methods, the success of our detection approach is heavily dependent on the adequate representation of training and validation data. Thus, the main questions regarding the appropriate modeling of industrial network traffic shall be answered by the help of a set of experiments. One objective of the experiments is to find an adequate representation of *sequences* of packets. The smallest context of a network communication can be modeled by a previous and a next packet. Since in other machine learning applications 4-grams have outperformed 3-grams [14], likewise the modeling of sequences of four packets may result in better detection accuracy than learning sequences of three packets. Another experimental objective is the modeling of communication relations between network devices. Usually, a network's traffic involves numerous of such communication relations. According to the idea of flow analysis in the field of network security, one can assume that a communication-specific mapping of packet sequences allows a more precise modeling of the traffic seen. The experiments shall help to measure a positive effect, if present. Finally, two remaining techniques to distinguish are time-independent versus temporal mapping. Whereas packet sequences can be mapped to n-packets vectors without any time constraints, temporal mappings only consider packet sequences whose duration (i.e., timestamp of first packet to timestamp of last packet) is within a certain time limit. To sum up, the main three questions answered in this paper are the following:

- Does learning sequences with different network packet counts result in a better/worse detection capability?
- Is the detection capability higher if packet sequences are learned separately for each communication relation or is a learning among all relations sufficient?
- Is there a significant difference in learning the sequences with and without temporal limits?

III. Datasets

The network traffic we have used for our experiments is taken from three network segments of two industrial sites. Each network segment consists of human-machine interfaces (HMI), programmable logic controllers (PLC), and peripheral devices cooperating for monitoring and controlling a complex and permanent automation process. The three datasets have the following characteristics:

`MoveA`. The first trace of network data origins from a movable system consisting of 104 network devices. [1]

`MoveB`. The second trace is taken from another segment of the same industrial site. This is similar in size but has, compared to the first segment, only half the packet rate, i.e., the number of packets transmitted within a second. Both traces serve as examples for complex industrial network segments that significantly differ in the packet rate.

`Carrier`. The last dataset is taken from a system which is, compared to the previous traces, very different in size (18 network devices) and purpose. The source is a flexible lignite conveyer system situated in an open-face mine used for bridging the way between the excavator and long-distance vehicles transporting the lignite out of the mine. That trace is used to show that a self-configuring anomaly detection is suitable even for small industrial networks.

For a consistent analysis the same industrial protocol is monitored in each trace, namely Profinet IO. This protocol serves as an example for a standardized, Ethernet-based industrial control protocol that is the first choice for important automation vendors. Another argument for using this protocol is the existence of known vulnerabilities and protocol-level attacks, such as Man-in-the-Middle and Denial-of-Service attacks [15], [16], [17]. These can be used for future evaluation of the detection approach towards real-world attacks. The characteristics of the Profinet IO (PN) traffic in the datasets are listed in Table I.

TABLE I: Characteristics of the datasets

	`MoveA`	`MoveB`	`Carrier`
size [MB]	6,765	2,800	3,300
duration [min]	98.56	40.13	47.27
# PN packets [10^3]	70,988	14,890	13,959
# PN packet rate [pkts/s]	12,005	6,186	4,922
# PN devices	104	107	18
# PN comm. relations	111	107	18
# PN message types	5	5	4

IV. Methodology

This section explains the mapping types applied for transferring network data to feature vectors, the criteria used to evaluate the learning success, and the composition of the validation sets.

A. Mapping Types

For answering the questions posed in Section II-C, each dataset is mapped to OCSVM feature vectors in multiple ways. Each mapping is characterized by the choice of three *mapping attributes*: (1) the number of packets mapped together; (2) whether communication relations are regarded; (3) the applied time limit while mapping, if defined. These attributes with chosen values are explained in the following.

1) 3-Packets Vectors versus 4-Packets Vectors: As motivated in Section II-C, sequences of three and four packets are investigated. A vector describing a sequence of n packets is built by concatenating n subvectors built from the n single network packets. The mapped data of each network packet are the addresses of the sending and the receiving device and the Profinet IO message type. Defining a maximum number of devices and by matching addresses to device IDs, all packet data can be treated as categorical features. Thus, each of the three packet attributes can be mapped to a numerical representation by constructing a k-dimensional vector v whose values at position $i \in \{1 \ldots k\}$ are defined as

$$v(i) = \begin{cases} 1, & \text{feature value is of the } i\text{-th category} \\ 0, & \text{otherwise} \end{cases} . \quad (1)$$

Given a maximum number of $d \in \mathbb{N}$ network devices in the monitored segment and $t \in \mathbb{N}$ different Profinet IO message types, a sequence of $n \in \mathbb{N}$ packets, each described by the 3-tuple (source device ID, destination device ID, message type ID) is mapped to a binary feature vector of dimension $n \times (2d + t) \in \mathbb{N}$. In terms of dataset `MoveA`, for instance, with 104 network devices and 5 different message types 4-packets feature vectors require a dimension of $4 \times (2 \times 104 + 5) = 852$. This kind of vector construction takes both the packet content and the packet order into account. More information of mapping categorical features can be found in [18].

2) Global Vectors versus Communication Vectors: The decision whether packet sequences are considered separately for each communication relation determines the choice of the context that is mapped with each network packet. Without distinction of communication relations, each n-packets vector is built from the last n packets seen in the network segment without regard to the source or destination device of the

[1] Due to demands of the vendors and network owners we cannot provide more information here. Since the abstract automation task has low influence on the traffic characteristics at protocol level in general, it is also not relevant for the results of this paper.

packets. These n-packets vectors describe packet sequences across all communication relations and are further referred to as *global vectors*. If packet sequences are mapped separately for each communication relation, in contrast, each n-packets vector is built from the last n packets exchanged between the respective pair of network devices. This kind of vectors are onwards named *communication vectors*. Investigating the global mapping is eligible, because industrial communication on field level is highly cyclic with mainly deterministic protocol behaviour among predefined sets of devices compared to communication on higher industrial levels or standard IT networks. In contrast, communication vectors may better model the individual characteristics of communication relations.

3) Time-independent Vectors versus Period Vectors: Except for the packet order, the construction of multi-packets vectors does not take any temporal characteristics into account. The idea of enriching the training of OCSVM models by temporal traffic aspects led to the idea of period vectors. A period vector contains information about all packet sequences seen in a defined interval, e.g., three seconds. Given the set $S = \{v_1, v_2, \dots v_m\}, m \in \mathbb{N}$ of n-packets vectors derived from all the sequences of n packets seen while analyzing a dataset, each period vector p is of dimension $|S| = m$ and each position $i \in \{1 \dots m\}$ is defined as follows:

$$p(i) = \begin{cases} 1, & \text{vector } v_i \text{ has been seen in the current period} \\ 0, & \text{otherwise} \end{cases} \qquad (2)$$

The intervals for composing period vectors are one and three seconds. The aim is to find out whether the use of OCSVM with period vectors results in better models and whether the choice of the period duration significantly influences the learning success. The combination of these criteria for each trace results in twelve *mapping types* applied to the network data to build feature vectors. For example, one mapping constructs period vectors by mapping all communication-specific 3-packets vectors seen in an interval of three seconds, another mapping just maps all sequences of four packets across all communication relations and without any time limits.

B. Evaluation Criteria

In order to evaluate the quality of the OCSVM models constructed by the several mapping types we measured various criteria. These are explained in the following by use of the following terms. The *training set* (T) consists of 50% of the vectors constructed from the respective normal dataset and is used for training the OCSVM model. The *validation set* (V) is a set with the other half of normal vectors complemented by vectors derived from the related anomalous dataset.

1) Cross Validation Rate: The measure of k-fold cross validation denotes the partitioning of a training set into $k \in \mathbb{N}$ complementary subsets of an equal number of vectors and k rounds of validating each subset by an OCSVM model that has been trained beforehand using the $k - 1$ other subsets. The resulting cross validation rate is the mean of all k results of the k cross validation rounds. The cross validation rate is a procedure to assess the ability of a model to extrapolate from the training set to unseen data without the need of a validation set. It answers the following question: Can the mapping type applied to the respective dataset produce vectors that prevent an overfitting of the OCSVM model while learning?

2) True-negative Rate: In our context of network intrusion detection the focus is on finding intrusion instances, i.e., anomalous network data. Consequently, anomalous data is referred to as *positives* while normal network data is interpreted as *negatives*. The true-negative rate (n_t), the ratio of normal vectors that are classified as such by the learned model to the number of all normal vectors, are determined for both the training and the validation set. This measure has to be sufficiently high, because the higher the true-negative rate the lower the false-positive rate $p_f = 1 - n_t$ and so the lower the rate of false alarms for our intrusion detection approach.

3) True-positive Rate (Recall): This measure is determined for the validation set only and represents the ratio of anomalous vectors that are classified as anomalous by the learned model to the number of all indeed anomalous vectors in the validation set. This measure is associated to the concept of *false negatives*, i.e., anomalous packet sequences that are classified as normal using the OCSVM model: the higher the recall the lower the false-negative rate and so the amount of undetected anomalous traffic.

Based on these basic numbers, we finally calculated the more complex measures *precision* and *balanced f-score* for each model in order to identify the best models.

C. Validation Sets

Each validation set is a composition of half of the normal vectors derived from a normal dataset and all anomalous vectors derived from anomalous traffic. A drawback of using real-world network traffic from productive industrial installations is the lack of possibilities to produce malicious network data, such as packets containing anomalous data or Man-in-the-Middle and Denial-of-Service attacks. Especially in highly tailored industrial sites, like the data sources of this work, slightly altered traffic can cause the outage of industrial devices that may lead to physical damages in the network.

While for dataset `MoveB` the operators explicitly performed unusual actions to provoke anomalous network traffic, the datasets `MoveA` and `Carrier` do not contain anomalous traffic. Here, anomalous vectors are derived from foreign traffic. Due to the equal device number and purpose of the source segments of `MoveA` and `MoveB`, the OCSVM model learned from dataset `MoveA` is validated with the help of all vectors derived from `MoveB`, that have not been constructed before from the normal dataset. (If two networks N_1 and N_2 use a similar local address space, it is possible that a packet sequence from network N_1 results in the same vector as a packet sequence from network N_2. The applied procedure eliminates these vectors from the designated anomalous vector set V_2 constructed from N_2, since as part of the normal dataset N_1 these vectors are already defined as normal.) In the same way, the model constructed from the dataset `Carrier` is tested with data from a foreign, completely different installation.

We argue that this procedure is legitimate to assess the learnability of normal traffic in industrial sites. If the ability is high to recognize foreign traffic (packets and packet sequences) as not normal for the local network, this also indicates a good anomaly detection capability (recall) of the learned model regarding the attacks presented in [17], especially if this meets a high true-negative rate and precision of the model.

V. Experimental Results

In this section, we describe the composition of our experiments and discuss the results of the application of several learned OCSVM models to the datasets.

A. Experimental Procedure

We set up three main experiments, each representing a different scenario. In each experiment one of the datasets introduced in Section III is used as normal industrial network traffic for deriving a set of normal vectors (N) for the training and validation set. The validation set is enriched with a set of anomalous vectors (A) derived from another dataset, as explained in Section IV-C.

Experiment 1: In the first experiment, the set of normal vectors N is derived from `MoveA`. For the generation of the set of anomalous vectors A the dataset `MoveB` is used, that is taken from the same industrial installation but from a different network segment. The traffic of both segments mainly differ in the packet rate. This experiment is to demonstrate how well strongly related network traffic can be recognized as foreign traffic in a monitored network segment.

Experiment 2: This experiment takes one part of `MoveB` as normal traffic for composing set N and another part of `MoveB` containing anomalous network traffic from the same segment for derivation of set A. Hence, this setting is the one with the most fine-grained anomalies. It serves as example for a scenario, in which the success of the anomaly detection is measured on network data from the same segment with identical, but anomalously operating devices.

Experiment 3: In the last experiment, set N is derived from dataset `Carrier`. The set of anomalous vectors is generated from a simulated PROFINET IO network of equal size and similar PROFINET IO cycle time. Aim of this experiment is to prove that a self-configuring anomaly detection using OCSVM is also applicable for small industrial networks.

We applied the mapping types, explained in Section IV-A, to the datasets of each experiment. In each case, this led to twelve different sets of training and test vectors: four sets of time-independent multi-packets vectors (communication-specific as well as global 3- and 4-packets vectors), four sets of period vectors for a period of three seconds and another four sets for a period of one second. The resulting numbers of OCSVM vectors can be found in Table II. As described in Section IV-C, half of the normal OCSVM feature vectors are used for training and the other half for validation together with all anomalous vectors.

In order to perform a systematic search for an expressive model in each experiment, the vectors of each mapping have been used to train various OCSVM models. We used four different kernels: linear, polynomial, radial basis function (RBF), and sigmoid with kernel parameters $\gamma = ((1/n), 2^{-15}, 2^{-13}, \dots 2^{3})$ with n as the number of training vectors and $\nu = (0.01, 0.02 \dots 0.09, 0.1, 0.2 \dots 0.9, 1)$. This resulted in 646 different OCSVM models tested in each experiment for each of the twelve mapping types. Next, the cross validation rates for two and five complementary subsets of the training set, the true-negative rates for the training and validation set as well as recall, precision, and balanced f-score have been calculated for every model (cf. Section IV-B).

B. Discussion

The application of the described mapping types on a high number of monitored packets leads in general to a comparatively very low number of *different* OCSVM feature vectors (cf. Table II). This fact proves the expected homogeneity of industrial traffic that can be explained as follows: Compared to standard IT networks, industrial networks are characterized by a strict setup of devices, communication relations, and data exchanged. This leads to constantly recurring packet sequences between devices transferring data in a well-defined range. Hence, a small number of different packet sequences is seen compared to the number of packets in each dataset. For a communication relation, the homogeneity is particularly high because mapping the datasets to communication vectors generally produces even smaller sets of vectors compared to the global mapping. In case of experiment 1, the number of global 3-packets vectors is higher by factor 232 and the number of 4-packets vectors by factor 317 than the communication vectors. From the homogeneity of the traffic traces is concluded that a longer training over larger datasets would be unlikely to lead to better models, i.e., the necessary duration of a training phase is usually well bounded by the homogenous nature of the traffic. Due to the construction of period vectors (cf. Section IV-A), the dimension of each period vector is the total number (Σ) of the n-packets vectors of the mapping type.

The outcome of the experiments is summarized in Table III. For each experiment, the best model learned for each mapping is indicated. On the upper half of the table, the results for the direct mappings of three and four packets to feature vectors are listed, i.e., communication (*com 3* and *com 4*) and global vectors (*global 3* and *global 4*). On the bottom half, the results using period vectors summarizing the n-packets vectors seen in a time frame of three seconds are shown analogously. The presentation of the results of period vectors for one second is omitted since no noteworthy differences can be observed to the three-seconds equivalent. We define the best model as the one with a good precision as first priority and a comparatively good recall (both combined in the f-score), because a low false-positive rate is the absolute prerequisite for a real-word detector regardless of its quality in terms of recall. The provided cross validation rate is the minimum reached with two and five groups. The results of the best model are provided with information about the used OCSVM parameters. We discuss the results by means of the initial questions of Section II-C.

Length of Packet Sequences: The results show that the mapping of sequences of four packets (time-independent as well as period vectors) does not necessarily lead to better models, i.e., models with significantly higher precision and f-score. In one case (s–t), precision and recall are even about 6% respectively 11% lower for $n = 4$ than for the 3-packets equivalent. Thus, we probably will not further investigate in the mapping of four network packets. Instead, we will focus on the modelling of the minimal context of packets as 3-packets vectors in next refinements of our protocol-specific detection.

Benefit of Communication-specific Learning: Noteworthy differences between global and communication-specific mappings can only be observed for experiment 2 (e–h,q–t).

TABLE II: Vector sets

	experiment 1			experiment 2			experiment 3		
	normal	anomal.	Σ	normal	anomal.	Σ	normal	anomal.	Σ
# packets	71573687	29785627	101359314	14950136	14835491	29785627	14112831	3691633	17804464
# PN packets	70988226	29668927	100657153	14890492	14778435	29668927	13959719	3599837	17559556
n-packets vectors									
com 3	3998	584	4582	810	1853	2663	393	157	550
com 4	7218	1280	8498	1592	2505	4097	619	215	834
global 3	920469	146449	1066918	154322	120813	275135	11415	757	12172
global 4	2236125	463113	2699238	271129	243729	514858	39907	1602	41509
period vectors (3 seconds)									
com 3	1860	580	2440	284	298	582	846	38	884
com 4	1864	588	2452	290	299	589	871	69	940
global 3	1869	671	2540	321	352	673	929	508	1437
global 4	1869	673	2542	321	352	673	929	529	1458
period vectors (1 second)									
com 3	5278	888	6166	471	419	890	1199	46	1245
com 4	5303	853	6156	448	406	854	1286	89	1375
global 3	5327	1921	7248	942	980	1922	2821	1233	4054
global 4	5326	1920	7246	942	979	1921	2821	1373	4194

TABLE III: Experimental results

				training set		validation set		model parameters			
experiment	mapping	%precision	%f-score	%crossval	%true neg.	%true neg.	%recall	kernel	ν	γ	
n-packets vectors											
1	com 3	75.05	44.55	89.55	90.30	88.64	31.68	sigmoid	0.1	0.5	(a)
	com 4	73.62	42.87	70.13	99.72	78.61	30.23	polynomial	0.04	8	(b)
	global 3	69.45	33.42	90.06	90.38	90.26	22.00	polynomial	0.1	0.01	(c)
	global 4	69.25	34.45	85.20	92.15	87.48	22.93	RBF	0.01	0.5	(d)
2	com 3	91.23	94.89	81.73	94.57	86.42	98.87	sigmoid	0.01	0.125	(e)
	com 4	91.71	95.36	87.56	95.23	86.81	99.32	sigmoid	0.1	0.125	(f)
	global 3	61.53	64.97	35.97	72.46	41.51	68.81	RBF	0.03	0.5	(g)
	global 4	61.40	64.98	35.32	71.81	41.45	69.00	RBF	0.3	0.5	(h)
3	com 3	99.74	97.93	94.92	99.49	100	96.18	sigmoid	0.01	2	(i)
	com 4	99.04	99.52	96.45	98.71	99.35	100	sigmoid	0.03	2	(j)
	global 3	99.20	99.60	98.79	99.19	99.19	100	sigmoid	0.01	2	(k)
	global 4	99.09	99.54	99.09	98.98	99.18	100	sigmoid	0.01	0.125	(l)
period vectors (3 seconds)											
1	com 3	98.83	99.41	98.17	99.25	98.39	100	RBF	0.01	$1/\|T\|$	(m)
	com 4	98.52	99.26	98.07	98.28	98.71	100	RBF	0.01	2^{-11}	(n)
	global 3	98.32	99.15	97.97	97.97	98.61	100	sigmoid	0.02	2^{-5}	(o)
	global 4	98.31	99.08	95.62	98.29	98.29	99.85	sigmoid	0.01	2^{-7}	(p)
2	com 3	85.12	58.27	58.45	100	84.51	44.30	polynomial	0.02	8	(q)
	com 4	86.58	58.77	66.21	99.31	86.90	44.48	polynomial	0.04	2^{-5}	(r)
	global 3	82.53	46.52	77.64	93.17	93.13	32.39	sigmoid	0.03	2^{-7}	(s)
	global 4	76.51	33.33	82.61	93.17	93.75	21.31	sigmoid	0.02	2^{-7}	(t)
3	com 3	97.69	98.83	92.91	98.35	96.93	100	RBF	0.01	2^{-13}	(u)
	com 4	96.56	98.25	94.50	97.48	95.40	100	RBF	0.01	2^{-13}	(v)
	global 3	99.36	99.68	98.93	98.71	100	100	sigmoid	0.01	2^{-5}	(w)
	global 4	99.25	99.63	99.79	98.93	99.57	100	sigmoid	0.01	0.125	(x)

Here, the precision, recall and f-score for time-independent communication-specific learning (e,f) are each about 30% higher than for global learning (g,h). Concerning the period vectors of experiment 2, the difference is less and most obvious for recall and f-score for four packets (r,t) being 23-25% higher for the communication-specific vectors. In the other two experiments, communication-specific and global learning lead to similar results. Here, a distinction on communication level does not enhance the quality of the learned model. Only the less number of communication vectors may be an advantage.

Temporal Learning: In experiment 1, in which the training and the validation dataset mainly differ in the packet rate, period vectors clearly outperform the time-independent multi-packets vectors in all measured criteria. Thus, period vectors seem to be powerful for detecting attacks that lead to a change in the number of different packet sequences seen in a time frame. We will further investigate this issue. The choice of one or three seconds as a period has no observable effect to the quality of the found models. The less number of three-seconds period vectors compared to the one-second equivalent militates for the application of the longer period.

OCSVM Configuration: It is obvious that three of the four tested kernel types provide the best model for at least four mappings. The linear kernel is not present. In over half of the cases (13/24), a model constructed with a sigmoid kernel is the best. However, one can also conclude that if the kernel parameters are well chosen, OCSVM-based anomaly detection in industrial networks can work with several kernels. For our monitoring data, i.e., sequences of packets with source and destination address and the PROFINET IO message type, the choice of the OCSVM kernel obviously plays a minor role for finding models of good quality. Mapping further payload data in future investigations in order to monitor application data may emerge the explicit suitability of a kernel.

In general, the quality of the constructed OCSVM models (expressed by the balanced f-score) is in 12 of 24 cases over 97%. In ten of these cases the recall is 100% with over 96% precision and a true-negative rate of more than 95%. We conclude from the combination of these high values that a model is found with high detection accuracy not being overfitted to the training data and that is at the same time capable to recognize the training data itself as normal (true-negative rate for the training set). This fact is crucial for industrial networks because a good capability to reason from learned normal traffic to unseen normal traffic is worth little if normal traffic of the training phase is not recognized as normal anymore in the detection phase (indicated by a low true-negative rate for the training set). We consider the mentioned twelve cases as evidence of the good learnability of industrial traffic for a self-configuring anomaly detection using OCSVM. Nevertheless, in industrial networks with huge packet rates, the missing of only 1% in the true-negative rate can result in an unacceptably high number of false positives (false alarms). Here, combining multiple models by a voting algorithm could further reduce false-positive rates.

VI. RELATED WORK

Due to their homogenous traffic, industrial control networks are predestined for the use of anomaly detection methods, which is subject of a few related approaches. They can be categorized based on the source of data used for feature extraction: process-based, packet-based, and hybrid approaches.

Process-based anomaly detection does not analyze packet data and relies only on process data. The detection process in [19] applies pattern matching. It combines *Autoassociative Kernel Regression* for model generation with a binary hypothesis technique, called *Sequential Probability Ratio Test*, during detection. The proposed kind of model generation, however, relies on the assumption that security violations are reflected by a change in the system usage, which is subject of the monitoring. This obviously limits the detection capability. In [20] and [21] contributions to a state-based IDS are presented. The system performs anomaly detection based on a decision whether the monitored system enters a critical state. For this purpose, a central virtual image of the physical state of the whole system is set up and periodically updated. Another approach [22] also considers process data. Here, *fuzzy C means* are used to find process data anomalies caused by sensor failures, communication interruption, or storage exception.

Packet-based approaches rely on packet data. The authors of [23] also focus on n-gram anomaly detection, in which each gram refers to the attribute extraction from a network packet. In [24] a back-propagation algorithm is used to build the neural network for a network-based intrusion detection system. Although these approaches provide relevant contributions, both are based on supervised learning that depends on labeled input data, i.e., requires normal as well as attack data. A further machine learning algorithm (*Self-Organizing Maps*) is used in [25]. Although application-payload-based features are seen to be an accurate mean to monitor the sanity of industrial environments [26], the approach only uses features extracted from traffic up to the transport layer (e.g., number of TCP and UDP connections, duration of connections, sent and received amount of data). The multi-agent IDS presented in [27] also relies on transport layer features. Unfortunately, general datasets not referring to industrial traffic are used for evaluating the detection accuracy.

Hybrid approaches use packet data and other process or system data. The algorithm in [28] uses deep packet inspection and estimates if a network packet has anomalous effect on a memory variable of an industrial device. This approach, however, requires both detailed understanding of the industrial protocol and extensive knowledge about the RAM variables of all PLC of the ICS. The approach presented in [29] uses *Dempster-Shafer's Theory of Evidence* for data-fusion-based anomaly detection. The authors aim to extract features from different sources of information, such as control hardware and physical sensors as well as signature-based and other intrusion detection programs. Thus, an operation seems at first level of detection to rely on data provided by extern sensors.

All of the aforementioned approaches are to some extent highly process-specific, but in the same time leave out any details about the used application protocols. Malware, such as Stuxnet [1], indicates that attacks on industrial environments are *targeted*, i.e., specially tailored to the target system. We doubt that the suggested methods can detect misuse of application protocols as described for PROFINET IO [15], [16], [17] or Modbus [30], [31], [32].

VII. Conclusion and Future Work

The use of machine learning techniques for monitoring industrial installations is a very promising field. Many papers only discuss the application of methods. However, learning methods have to be evaluated on real-world data from early stage in order to successfully identify suitable methods. We focus on the development of a self-configuring, network-based and protocol-specific anomaly detection for industrial environments. Since we have identified OCSVM as a promising learning method, we measured in this work the quality of numerous OCSVM models trained by various mappings of multiple real-world industrial traffic traces. In contrast to existing methods, we discussed and evaluated different methods of mapping packet sequences of industrial traffic to machine learning vectors. Although the monitored packet data is limited to a few data of the example industrial protocol, OCSVM models with high quality in terms of balanced f-score, precision, and recall have been identified. After these promising preliminary results of our protocol-specific anomaly detection approach, we plan to extend the learned features by further payload information in order to guarantee an acceptable false-positive rate even in industrial sites with high data load.

Acknowledgment

The authors gratefully acknowledge funding from the German Federal Ministry of Education and Research (BMBF) via the project INDI (funding reference number: 16KIS0156).

References

[1] N. Falliere, L. O. Murchu, and E. Chien, "W32.Stuxnet Dossier, Version 1.4," Cupertino, 2011.

[2] D. Hadziosmanović, L. Simionato, D. Bolzoni, E. Zambon, and S. Etalle, "N-gram Against the Machine: On the Feasibility of the N-gram Network Analysis for Binary Protocols," in *Proc. of the 15th International Symposium on Research in Attacks, Intrusions and Defenses (RAID)*, ser. LNCS, vol. 7462. Springer, Heidelberg, 2012, pp. 354–373.

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

[4] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford University Press Inc., Oxford, 1995.

[5] N. Ullman, *Elementary Statistics: An Applied Approach*. John Wiley Sons, New York, 1978.

[6] N. Japkowicz, C. Myers, and M. Gluck, "A Novelty Detection Approach to Classification," in *Proc. of the 14th International Joint Conference on Artificial Intelligence (IJCAI) Vol. 1*. Morgan Kaufmann Publishers Inc., San Francisco, 1995, pp. 518–523.

[7] P. Baldi and K. Hornik, "Neural Networks and Principal Component Analysis: Learning from Examples Without Local Minima," *Neural Networks*, vol. 2, no. 1, pp. 53–58, 1989.

[8] T. Kohonen, "Self-organized Formation of Topologically Correct Feature Maps," *Biological Cybernetics*, vol. 43, no. 1, pp. 59–69, 1982.

[9] G. A. Carpenter, S. Grossberg, and D. B. Rosen, "ART 2-A: An Adaptive Resonance Algorithm for Rapid Category Learning and Recognition," *Neural Networks*, vol. 4, no. 4, pp. 493–504, 1991.

[10] A. Ypma and R. Duin, "Support Objects for Domain Approximation," in *ICANN 98*, ser. Perspectives in Neural Computing, L. Niklasson, M. Bodn, and T. Ziemke, Eds. Springer, London, 1998, pp. 719–724.

[11] D. M. Tax, "One-class Classification," Ph.D. dissertation, Delft University of Technology, 2001.

[12] Schölkopf, B., Williamson, R., Smola, A., and Shawe-Taylor, J., "SV Estimation of a Distributions Support," in *Proc. of Neural Information Processing Systems (NIPS)*. MIT Press, Cambridge, 1999, pp. 582–588.

[13] C.-C. Chang and C.-J. Lin. (2001) LIBSVM: A Library for Support Vector Machines. [Online]. Available: http://www.csie.ntu.edu.tw/∼cjlin/libsvm/

[14] C. Wressnegger, G. Schwenk, D. Arp, and K. Rieck, "A Close Look on *n*-Grams in Intrusion Detection: Anomaly Detection vs. Classification," in *Proc. of the 2013 ACM Workshop on Artificial Intelligence and Security (AISec*. ACM, New york, 2013.

[15] M. Baud and M. Felser, "PROFINET IO Device Emulator based on the Man-in-the-middle Attack," in *Proc. of the 11th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE Press, New York, 2006, pp. 437–440.

[16] J. Åkerberg and M. Björkman, "Exploring Security in PROFINET IO," in *Proc. of the 33rd Annual IEEE International Computer Software and Applications Conference (COMPSAC)*. IEEE Press, New York, 2009, pp. 406–412.

[17] A. Paul, F. Schuster, and H. König, "Towards the Protection of Industrial Control Systems – Conclusions of a Vulnerability Analysis of PROFINET IO," in *Proc. of the 10th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA)*, ser. LNCS, vol. 7967. Springer, Heidelberg, 2013, pp. 160–176.

[18] F. Schuster, A. Paul, and H. König, "Towards Learning Normality for Anomaly Detection in Industrial Control Networks," in *Proc. of the 7th International Conference on Autonomous Infrastructure, Management, and Security (AIMS)*, ser. LNCS, vol. 7943. Springer, Heidelberg, 2013, pp. 61–72.

[19] D. Yang, A. Usynin, and J. W. Hines, "Anomaly-based Intrusion Detection for SCADA Systems," in *Proc. of the Fifth International Topical Meeting on Nuclear Plant Instrumentation, Control and Human Machine Interface Technologies*. Curran Associates, Red Hook, 2006, pp. 12–16.

[20] A. Carcano, I. N. Fovino, M. Masera, and A. Trombetta, "State-Based Network Intrusion Detection Systems for SCADA Protocols: A Proof of Concept," in *Proc. of the 4th International Workshop on Critical Information Infrastructures Security (CRITIS)*, ser. LNCS, vol. 6072. Springer, Heidelberg, 2009, pp. 138–150.

[21] A. Carcano, A. Coletta, M. Guglielmi, M. Masera, I. N. Fovino, and A. Trombetta, "A Multidimensional Critical State Analysis for Detecting Intrusions in SCADA Systems," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 2, pp. 179–186, 2011.

[22] J. Zhao, K. Liu, W. Wang, and Y. Liu, "Adaptive Fuzzy Clustering-based Anomaly Data Detection in Energy Systems of Steel Industry," in *Information Sciences*, vol. 259, 2014, pp. 335–345.

[23] O. Linda, T. Vollmer, and M. Manic, "Neural Network based Intrusion Detection System for Critical Infrastructures," in *Proc. of the 2009 International Joint Conference on Neural Networks (IJCNN)*. IEEE Press, New York, 2009, pp. 1827–1834.

[24] W. Gao, T. Morris, B. Reaves, and D. Richey, "On SCADA Control System Command and Response Injection and Intrusion Detection," in *Proc. of the 5th eCrime Researchers Summit*. IEEE Press, New York, 2010, pp. 1–9.

[25] M. Mantere, M. Sailio, and S. Noponen, "A Module for Anomaly Detection in ICS Networks," in *Proc. of the 3rd Int. Conference on High Confidence Networked Systems*. ACM, New York, 2014, pp. 49–56.

[26] M. Mantere, I. Uusitalo, M. Sailio, and S. Noponen, "Challenges of Machine Learning Based Monitoring for Industrial Control System Networks," in *Proc. of the 26th International Conference on Advanced Information Networking and Applications (AINA)*. IEEE Press, New York, 2012, pp. 968–972.

[27] C.-H. Tsang and S. Kwong, "Multi-Agent Intrusion Detection System in Industrial Network using Ant Colony Clustering Approach and Unsupervised Feature Extraction," in *Proc. of the International Conference on Industrial Technology (ICIT)*, 2005, pp. 51–56.

[28] J. Rrushi and K.-D. Kang, "Detecting Anomalies in Process Control Networks," in *Critical Infrastructure Protection III*, ser. IFIP Advances in Information and Communication Technology (AICT), C. Palmer and S. Shenoi, Eds. Springer, 2009, vol. 311.

[29] B. Genge, C. Siaterlis, and G. Karopoulos, "Data Fusion-Based Anomaly Detection in Networked Critical Infrastructures," in *Proc. of the 43th IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE Press, New York, 2013, pp. 1–8.

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

[31] T. H. Morris, B. A. Jones, R. B. Vaughn, and Y. S. Dandass, "Deterministic Intrusion Detection Rules for MODBUS Protocols," in *46th Hawaii International Conference on System Sciences (HICSS)*. IEEE Press, New York, 2013, pp. 1773–1781.

[32] W. Gao and T. H. Morris, "On Cyber Attacks and Signature Based Intrusion Detection for MODBUS Based Industrial Control Systems," *Journal of Digital Forensics, Security and Law*, vol. 9, no. 1, pp. 37–56, 2014.