Chapter 11

# DETECTING ANOMALIES IN PROCESS CONTROL NETWORKS

Julian Rrushi and Kyoung-Don Kang

**Abstract** This paper presents the estimation-inspection algorithm, a statistical algorithm for anomaly detection in process control networks. The algorithm determines if the payload of a network packet that is about to be processed by a control system is normal or abnormal based on the effect that the packet will have on a variable stored in control system memory. The estimation part of the algorithm uses logistic regression integrated with maximum likelihood estimation in an inductive machine learning process to estimate a series of statistical parameters; these parameters are used in conjunction with logistic regression formulas to form a probability mass function for each variable stored in control system memory. The inspection part of the algorithm uses the probability mass functions to estimate the normalcy probability of a specific value that a network packet writes to a variable. Experimental results demonstrate that the algorithm is very effective at detecting anomalies in process control networks.

**Keywords:** Distributed control systems, anomaly detection, applied statistics

## 1. Introduction

After decades of research, most of the physical processes underlying a system such as a nuclear power plant are known. If a physical system is operated in a digital (cyber) mode, as is the case of some Generation III, most Generation III+ and all Generation IV nuclear reactors, one can argue that, with the available knowledge in hand, we have a good definition of normalcy about the physical side of such a cyber-physical system. Because several behavior profiles of control systems and networks are induced by physical processes in the physical side, it is intuitively appealing to leverage the knowledge of normalcy in the physical side to obtain an assessment of normal behavior in the cyber side, and, thus, estimate the concept of normalcy for the entire cyber-physical system.

With this objective in mind, we conducted an observational study on an experimental cyber-physical system formed by a limited number of elements of a distributed control system [20] and simulated components of an advanced boiling water reactor (ABWR) [4]. The study involved the statistical analysis of the contents of random access memory (RAM) of a programmable logic controller (PLC) that contains control logic computation data, input data and output data, which we call "RAM variable memory." We discovered that the evolution of the values of logical and continuous variables stored in RAM variable memory follow specific flows that persist over time. This finding motivated our development of the estimation-inspection algorithm for anomaly detection.

The estimation-inspection algorithm probabilistically estimates the normal flows of values of logical or continuous variables stored in the RAM variable memory of PLCs and determines if a network packet is normal or abnormal by considering the specific evolution of values of a logical or continuous variable caused by the network packet. Experimental results using a simple testbed demonstrate that the algorithm is very effective at detecting anomalies in process control networks.

## 2. Related Research

This section discusses related research on intrusion and anomaly detection in process control networks. Cheung, *et al.* [3] have examined protocol-level models for intrusion detection in process control networks. The models employ a definition of normalcy for payloads of byte-oriented protocols such as Modbus [13] and are derived from the protocol specifications and implementation guides. Protocol-level models and the estimation-inspection algorithm both focus on the inspection of network packets. Protocol-level models search for violations related to function codes, exception codes, protocol identifiers and other attributes. They also examine cross-field relationships because a legitimate value of a field may depend on the value of another field. On the other hand, the proposed anomaly detection approach focuses entirely on data fields and uses applied statistics to assess their legitimacy.

Some researchers [17–19] have applied reactor mirage theory (RMT) as a deception-based intrusion detection technique for process control networks in nuclear power plants. RMT, which is based on signal detection theory [8, 12], uses continuous simulation [16] based on genuine control network traffic. The proposed approach differs from RMT in that it addresses situations where attackers target control systems attached to real, operational equipment.

The challenges involved in detecting attacks on control systems have been discussed by Cardenas, *et al.* [1] and by Naess, *et al.* [15]. The approach of Cardenas, *et al.* is based on an understanding of the interactions between the control system and the physical system. They model the behavior of a physical system as a linear dynamical system and use the model to determine the effects of control commands on the physical parameters of the system in question. Their approach assumes that an attack on a control system produces abnormal behavior in the physical system by having negative effects on the system

parameters; thus, they use sequential detection theory to detect the negative effects. Our approach models the interactions between a control system and a physical system in terms of the evolutions of values of logical and continuous variables stored in the RAM variable memory of the control system. A statistical estimation technique is used to obtain a series of parameters that are used with logistic regression formulas to form a probability mass function for each variable stored in control system memory. For a control command to be deemed normal, the network packet that conveys it should cause an evolution of values of a logical or continuous variable that is deemed to be normal by the probabilistic model.

Naess, *et al.* propose an intrusion detection approach that uses high-level application-based policies implemented at the middleware level. The misuse policies are based on attack signatures, procedural-based policies that use execution patterns of monitored components, and interval-based policies that look for anomalies in parameter values and method invocation frequencies. Procedural-based policies are not comparable with our statistical approach, nor are misuse policies and interval-based policies that deal with method invocation frequencies. Our research suggests that interval-based policies take into account the state of the physical system when setting parameter thresholds. Naess, *et al.* discuss maximum and minimum value policies that look for parameter values that lie outside the range of allowable values. For instance, if the allowed set point for the linear position of a control rod used to adjust the reactivity of a nuclear reactor core [21] should be an even value between 6 and 24, a maximum and minimum value policy would classify a set point of 24 as normal. However, if the value of reactivity is high, moving the control rod from a low value to a linear position of 24 is abnormal and possibly very dangerous.

The approach of Naess, *et al.* incorporates the delta value and maximum average policies, which are used to detect unexpected variations in parameter values over a short amount of time and excesses of maximum distance from a moving average for each measurement, respectively. A consideration of the state of a physical system would enable delta value and maximum average policies to produce corrective responses that are initiated by control systems upon an equipment fault or breakage. To our knowledge, such corrective responses often involve set points that cause large and abrupt changes to parameter values. Naess, *et al.* also use interval-based policies that employ cumulative distribution functions to detect rare values given a history of normally-distributed values. Thus, these policies compare the next value of a parameter with some number of previous values of the same parameter. In our experience, the next normal value of a parameter also depends on the current values of other parameters that characterize a physical system. In a nuclear power plant, for example, the next value of the position of a turbine bypass valve depends on the current value of the pressure in the reactor vessel. Our approach addresses this issue by considering the complete state of a physical system when estimating a probability distribution for the next value of a logical or continuous variable.

Naedele and Biderbost [14] describe an approach for reducing the security-related events that occur in a process control environment to quantitative metrics, which are understandable by system operators with limited computer security expertise. The quantitative metrics are presented to system operators during plant operation, enabling them to assess if an attack is taking place. Naedele and Biderbost's approach and the proposed approach both consider the dynamics of a digitally-controlled physical system when estimating the normalcy of network packets. However, the two approaches differ in how they estimate the normalcy of network packets. Naedele and Biderbost's approach uses humans for pattern matching while the proposed approach engages statistical estimation theory. We believe that the proposed approach is better because it supports real-time detection while the human analysis of events takes at least a few seconds. Consider, for example, a malicious network packet that opens a circuit breaker to desynchronize a power generator in the electrical power grid. While Naedele and Biderbost's approach would likely detect the malicious network packet only after circuit breaker has been opened, the proposed approach would detect the packet before it is processed by the control system that controls the circuit breaker.

## 3. Statistical Approach for Anomaly Detection

This section presents our statistical approach for anomaly detection. In particular, it describes the mathematical modeling and statistical parameter estimation techniques, and the estimation-inspection algorithm.

### 3.1 Mathematical Modeling

The RAM variables of control systems can be expressed as a matrix $W$ whose elements model logical or continuous variables that store process measurement data or actuator control data along with set points:

$$W = \begin{bmatrix} x_1 \; x_2 \; \cdots \\ \cdots \\ \cdots \; x_l \\ x_{l+1} \; x_{l+2} \; \cdots \\ \cdots \\ \cdots \; x_m \\ x_{m+1} \; x_{m+2} \; \cdots \\ \cdots \\ \cdots \; x_n \\ x_{n+1} \; x_{n+2} \; \cdots \\ \cdots \\ \cdots \; x_g \end{bmatrix}$$

The elements $x_1, x_2, \ldots, x_l$ model input register variables; the elements $x_{l+1}$, $x_{l+2}, \ldots, x_m$ model holding register variables; the elements $x_{m+1}, x_{m+2}, \ldots$, $x_n$ model discrete input variables; and the elements $x_{n+1}, x_{n+2}, \ldots, x_g$ model

coil variables. A control system can hold as many as 65,536 variables of each type. If $q$ is the number of control systems in a process control network, then $l = 65{,}536q$, $m = 2l$, $n = 3l$ and $g = 4l$. In a real-world control system, it may be the case that not all the input register variables, holding register variables, discrete input variables and coil variables are needed; consequently, not all of them are defined.

Logical variables and continuous variables in RAM variable memory, and, thus, the elements of matrix $W$ are mapped to process parameters (i.e., variables characterizing the operation of physical equipment and/or physical processes) according to specific schemes (i.e., cyber-physical mappings) that depend on the communication protocol being used. In some byte-oriented protocols (e.g., Modbus), cyber-physical mappings are defined *ad hoc* by control engineers and are applied during device configuration. Other protocols (e.g., IEC 61850 [6]) have the cyber-physical mappings defined in their specifications. Process parameters are related to each other by mathematical formulas based on the processes taking place in the physical side of the system.

A cyber-physical mapping associates the physical or chemical relations between process parameters with functional relations among logical variables and continuous variables in RAM variable memory, and, thus, with the functional relations among the elements of matrix $W$. The functional relations, in turn, determine the logical data and continuous data assigned to sensor or actuator variables during the controlled operation of a physical system. Thus, given a process in the physical side of a controlled system along with a cyber-physical mapping, a value assigned to an element of matrix $W$ can be explained by consulting a set of other elements of $W$ under the assumption that the analysis is being performed on a safe operation of the controlled physical system.

The fundamental thesis of this research is that for every possible combination of values of $W$ elements, including the current value of $W[i][j]$, $W[i][j]$ may take any one of its possible values with a probability that varies from 0 to 1. We refer to the probability in question as the "normalcy probability." A normal transition flow step occurs when $W[i][j]$ takes a value whose associated normalcy probability is non-zero. Thus, a network packet that is about to write to $W[i][j]$ is classified as normal if it causes a normal transition of the current value of $W[i][j]$, i.e., it writes a value to $W[i][j]$ whose associated normalcy probability is non-zero. In the statistical context, we refer to the elements of matrix $W$ as $W[i][j]$ and $x_1, x_2, \ldots, x_g$ when we treat them as dependent variables and exposure variables, respectively. The estimation-inspection algorithm, which is described later in this section, estimates the probability distribution of the values of $W[i][j]$ given $x_1, x_2, \ldots, x_g$ and checks that a network packet that writes to $W[i][j]$ conveys a value for $W[i][j]$ whose associated normalcy probability is non-zero.

The possible values of each $W[i][j]$ lie in $\{min\,(W[i][j]),\ min\,(W[i][j]) + 1, \ldots,\ min\,(W[i][j]) + h\}$, where $min\,(W[i][j]) + h = max\,(W[i][j])$. We use the term "possible value" because each logical variable, by definition, may assume the value 0 or 1, while each continuous variable takes values from a defined

interval that depends on the process parameter to which it is mapped. In a nuclear power plant, for example, the continuous variable mapped to the reactor vessel pressure may take values that vary from 0 psi at plant start-up to a maximum value of 1,000 psi when the plant is operating at 100% thermal power. Similarly, if the maximum synchronous speed of a two-pole AC induction motor is 1,500 rpm, then the applied voltage frequency, which is used control the actual rotational speed of the motor, may assume values from 0 Hz to 25 Hz. Note that $W[i][j]$ can take negative values because it is possible for process measurements and actuator control data to have negative values.

We use stochastic vectors to store the probability distributions of $W[i][j]$ values. These stochastic vectors are defined by:

$$V_{W[i][j]} = \{ \begin{bmatrix} p_0 \\ p_1 \\ . \\ . \\ . \\ p_h \end{bmatrix} \mid p_0 + p_1 + ... + p_h = 1 \} \quad (1)$$

Let $p_k = V_{W[i][j]}[k]$ be the normalcy probability that $W[i][j]$ takes the value $min\,(W[i][j]) + k$ where $k \in \{0, 1, \ldots, h\}$. Thus, $p_0 = V_{W[i][j]}[0]$ is the normalcy probability that $W[i][j]$ takes the value $min\,(W[i][j])$; $p_1 = V_{W[i][j]}[1]$ is the normalcy probability that $W[i][j]$ takes the value $min\,(W[i][j]) + 1$; and so on.

We use a probability mass function $\Gamma_{W[i][j]}$ to model the normal data transition flows that may potentially be followed by element $W[i][j]$. The probability mass function $\Gamma_{W[i][j]}$ is defined by:

$$\Gamma_{W[i][j]} : x_1 \times ... \times x_{l+1} \times \ldots \times x_{m+1} \times \ldots \times x_{n+1} \times \ldots \times x_g \rightarrow V_{W[i][j]} \quad (2)$$

The estimation part of the estimation-inspection algorithm uses logistic regression integrated with maximum likelihood estimation in an inductive machine learning process to estimate a series of statistical parameters. These statistical parameters in conjunction with logistic regression formulas form a practical definition of the probability mass function $\Gamma_{W[i][j]}$ for each $W[i][j]$. The inspection part of the estimation-inspection algorithm uses the probability mass function $\Gamma_{W[i][j]}$ to estimate the normalcy probability of a specific value $min\,(W[i][j]) + k$ that a network packet is about to write to $W[i][j]$.

## 3.2 Statistical Parameter Estimation

As described later, an element $W[i][j]$ may take any one of its possible values with a probability that depends on $x_1, x_2, \ldots, x_g$ and the statistical parameters $\alpha$(s) and $\beta$(s). The parameters $\alpha$(s) are intercept terms while $\beta$(s) are coefficient terms. We estimate the statistical parameters using applied logistic regression analysis integrated with maximum likelihood estimation [5, 9]. The first step is to run a model of the controlled physical system normally and without any

attacks. The values of the logical and continuous variables in control system RAM are recorded in a database as they evolve over time.

Next, we have different individuals run the model multiple times. Despite undergoing similar training and certification regimens, different nuclear reactor operators usually adjust process parameters in different ways to reach the desired operational states. Furthermore, process-related events may be handled differently, but are considered normal operations as long as the desired tasks are performed correctly. What is important is that the model be used to generate a sample of network packets that characterizes the population of network packets during normal operation of the controlled physical system.

For each program variable modeled by $W[i][j]$, we create a database view with rows of the form $\{\varphi(W[i][j]), x_1, x_2, ..., x_g\}$, where $\varphi(W[i][j])$ denotes the next value of $W[i][j]$. $\varphi(W[i][j])$ is extracted from a network packet transmitted over a process control network that is about to write $W[i][j]$, while the record of values of $x_1, x_2, \ldots, x_g$ is a snapshot of the current values of the elements of matrix $W$ just before the network packet changes the value of $W[i][j]$ to $\varphi(W[i][j])$.

We now consider the case where $W[i][j]$ models a continuous variable. If (in statistical terms) each possible value of $W[i][j]$ is considered to be an outcome category, then ordinal logistic regression is applicable because the categories (in general) are ordered in controlled physical systems. In a nuclear power plant, for example, the possible values of continuous variables mapped to process parameters (e.g., reactor vessel pressure, reactor water level, neutron population in the reactor core and steam flow rate) are ordered. In ordinal logistic regression, comparisons between the contiguous values of a dependent variable play a key role in estimating their probabilities of occurrence. Since the possible values of $W[i][j]$ lie in $[min\,(W[i][j]),\ min\,(W[i][j]) + h]$, there are $h$ possible comparisons between contiguous values of $W[i][j]$. Consequently, according to ordinal logistic regression, there are $h$ intercept terms $\alpha$ in the ordinal logistic model $\alpha_1, \alpha_2, ..., \alpha_h$.

An intercept term $\alpha_k$ is defined for each value $min\,(W[i][j]) + k$ of $W[i][j]$ such that $k \neq 0$. Later in this section we will see that $\alpha_k$ is used to estimate the probability that $W[i][j]$ takes the value $min\,(W[i][j]) + k$. We will also show that there is no $\alpha_0$ defined for $min\,(W[i][j])$. Since the logistic model under consideration is ordinal rather than polytomous, there is only one coefficient term $\beta_a$ associated with each exposure variable $x_a$ where $a \in \{1, 2, \ldots, g\}$. Furthermore, there is a unique set of coefficient terms $\beta_1, \beta_2, \ldots, \beta_g$ defined for all values $min\,(W[i][j])+k$ of $W[i][j]$. Like the intercept term $\alpha_k$, the coefficient terms $\beta_1, \beta_2, \ldots, \beta_g$ are also used to estimate the probability that $W[i][j]$ takes the value $min\,(W[i][j]) + k$.

Given $x_1, x_2, \ldots, x_g$, the probability that $W[i][j]$ takes a value greater than or equal to $min\,(W[i][j]) + k$ is:

$$P\left(\varphi(W[i][j]) \geq min\,(W[i][j]) + k \mid W\right) = \frac{1}{1 + e^{-\left(\alpha_k + \sum_{a=1}^{g} \beta_a x_a\right)}} \tag{3}$$

As a matter of fact, we are interested in $min\,(W[i][j]) + k \geq 1$ because $P\,(\varphi(W[i][j]) \geq 0 \mid W) = 1$. Similarly, the probability that $W[i][j]$ takes a value greater than or equal to $min\,(W[i][j]) + k + 1$ given $x_1, x_2, ..., x_g$ is:

$$P\,(\varphi(W[i][j]) \geq min\,(W[i][j]) + k + 1 \mid W) = \frac{1}{1 + e^{-\left(\alpha_{k+1} + \sum_{a=1}^{g} \beta_a x_a\right)}} \quad (4)$$

Equations (3) and (4) are used to derive the probability that $W[i][j]$ takes the value $min\,(W[i][j]) + k$ given $x_1, x_2, ..., x_g$. The probability is given by:

$$P(\varphi(W[i][j]) = min(W[i][j]) + k \mid W) = P(\varphi(W[i][j]) \geq min(W[i][j]) + k \mid W) - P(\varphi(W[i][j]) \geq min(W[i][j]) + k + 1 \mid W) \quad (5)$$

Upon substituting Equations (3) and (4) into Equation (5), we obtain:

$$P\,(\varphi(W[i][j]) = min\,(W[i][j]) + k \mid W) = \frac{1}{1 + e^{-\left(\alpha_k + \sum_{a=1}^{g} \beta_a x_a\right)}} - \frac{1}{1 + e^{-\left(\alpha_{k+1} + \sum_{a=1}^{g} \beta_a x_a\right)}} \quad (6)$$

For the case where $k = 0$ and the value of $W[i][j]$ whose probability of occurrence is being estimated is $min\,(W[i][j])$, the minuend in Equation (6) is 1 because $P\,(\varphi(W[i][j]) \geq min(W[i][j]) \mid W) = 1$. This explains why no $\alpha_0$ is defined for $min\,(W[i][j])$ (i.e., when $k = 0$).

Next, we discuss the development of the likelihood function $L_{W[i][j]}$ for an element $W[i][j]$. The function $L_{W[i][j]}$ represents the joint probability for the likelihood of observing the data of the $d$ rows in the database view. Assuming that the rows of the database view are numbered from 1 to $d$, let $y_{bk}$ be an indicator variable defined on the $b^{\text{th}}$ row as follows:

$$y_{bk} = \begin{cases} 1 & \text{if in the } b^{\text{th}} \text{ row, } \varphi(W[i][j]) = min\,(W[i][j]) + k \\ 0 & \text{if in the } b^{\text{th}} \text{ row, } \varphi(W[i][j]) \neq min\,(W[i][j]) + k \end{cases} \quad (7)$$

The joint probability for the likelihood of observing the data in the database view is:

$$\prod_{b=1}^{d} \prod_{k=0}^{h} P\,(\varphi(W[i][j]) = min\,(W[i][j]) + k \mid W)^{y_{bk}} \quad (8)$$

Equation (8) estimates the individual contribution made by each row to the probability that $\varphi(W[i][j])$ is $min\,(W[i][j]) + k$, and then combines the individual likelihood contributions made by each row. Clearly, each row contributes the probability of one value $min\,(W[i][j]) + k$ taken by $W[i][j]$ because only one of the indicator variables is equal to 1. Upon substituting Equation (6) into Equation (8), we obtain:

$$\prod_{b=1}^{d} \prod_{k=0}^{h} \left( \frac{1}{1 + e^{-\left(\alpha_k + \sum_{a=1}^{g} \beta_a x_a\right)}} - \frac{1}{1 + e^{-\left(\alpha_{k+1} + \sum_{a=1}^{g} \beta_a x_a\right)}} \right)^{y_{bk}} \quad (9)$$

The values of the exposure variables $x_1, x_2, \ldots, x_g$ in Equation (9) are available from the database view because each individual row is processed by the equation. Therefore, after performing the multiplications of the probabilities contributed by each individual row, the likelihood function $L_{W[i][j]}$ appears as a function of the statistical parameters, and is given by:

$$L_{W[i][j]}\left(\alpha_1, \alpha_2, \ldots, \alpha_h, \beta_1, \beta_2, \ldots, \beta_g\right) \tag{10}$$

We estimate the values of the statistical parameters $\alpha_1, \alpha_2, \ldots, \alpha_h, \beta_1, \beta_2, \ldots, \beta_g$ that maximize $L_{W[i][j]}$ using the maximum likelihood technique [11]. We organize the parameters of the likelihood function $L_{W[i][j]}$ as a vector $\theta = (\theta_1, \theta_2, \ldots, \theta_{h+g})$. Maximizing $L_{W[i][j]}(\theta)$ is equivalent to maximizing $ln\left[L_{W[i][j]}(\theta)\right]$. If $r \in \{1, 2, ..., h+g\}$ and $\theta_r$ is the $r^{\text{th}}$ element of vector $\theta$, the values of the statistical parameters that maximize $L_{W[i][j]}(\theta)$ are the solutions of a system of equations of the form:

$$\frac{\partial ln\left[L_{W[i][j]}(\theta)\right]}{\partial \theta_r} = 0 \tag{11}$$

where the fraction is a partial derivative of the natural logarithm of the likelihood function $L_{W[i][j]}$ with respect to $\theta_r$. The solutions of the system of equations yield estimates of the parameters $\alpha_1, \alpha_2, \ldots, \alpha_h, \beta_1, \beta_2, \ldots, \beta_g$. Armed with the estimated values of the parameters, we return to Equation (6) and estimate the probability that $W[i][j]$ takes the value $min\left(W[i][j]\right) + k$ given the current values of the elements of matrix $W$. This is an integral component of the estimation-inspection algorithm, which is presented below.

Estimating $p_k = \left(P\left(\varphi(W[i][j]) = min\left(W[i][j]\right) + k \mid W\right)\right)$ and storing $p_k$ in $V_{W[i][j]}[k]$, for each $k \in \{0, 1, \ldots, h\}$, fills all the positions of stochastic vector $V_{W[i][j]}$. Iterating this procedure over every possible tuple of values of exposure variables $x_1, x_2, \ldots, x_g$ associates each tuple with a stochastic vector $V_{W[i][j]}$, which leads to the computation of the probability mass function $\Gamma_{W[i][j]}$.

We now consider the case where $W[i][j]$ models a logical variable that takes a value of 0 or 1. Since a logical variable matches the definition of a dichotomous measure in a statistical context, dichotomous logistic regression can be applied. In a dichotomous logistic model, there is only one intercept term $\alpha$ defined for the two possible values of $W[i][j]$, and only one coefficient term $\beta_a$ associated with each exposure variable $x_a$ where $a \in \{1, 2, \ldots, g\}$. Furthermore, a unique set of coefficient terms $\beta_1, \beta_2, \ldots, \beta_g$ is defined for the two possible values of $W[i][j]$. Upon applying the logistic function of the dichotomous logistic model, we obtain the probability that an element $W[i][j]$ takes the value 1:

$$P\left(\varphi(W[i][j]) = 1 | W\right) = \frac{1}{1 + e^{-\left(\alpha + \sum_{a=1}^{g} \beta_a x_a\right)}} \tag{12}$$

The probability that $W[i][j]$ takes the value 0 is given by:

$$P\left(\varphi(W[i][j]) = 0 | W\right) = 1 - P\left(\varphi(W[i][j]) = 1 | W\right) = 1 - \frac{1}{1 + e^{-\left(\alpha + \sum_{a=1}^{g} \beta_a x_a\right)}} \tag{13}$$

We arrange the rows of the database view so that for the first $c$ rows: $\varphi(W[i][j]) = 1$, and for the remaining $d - c$ rows: $\varphi(W[i][j]) = 0$. Let $P(X_b)$ denote $P(\varphi(W[i][j]) = 1 \mid W)$ for the $b^{\text{th}}$ row. Also in a dichotomous logistic model, the joint probability for the likelihood of observing the data in the database view is given by the likelihood function $L_{W[i][j]}$ defined by:

$$\prod_{b=1}^{c} P(X_b) \prod_{b=c+1}^{d} 1 - P(X_b) \tag{14}$$

Equation (14) estimates the individual likelihood contribution made by each row numbered from 1 to $c$ to the probability that $W[i][j]$ takes the value 1, along with the individual likelihood contribution made by each row numbered from $c+1$ to $d$ to the probability that $W[i][j]$ takes the value 0; it then combines the individual likelihood contributions made by each row. The values of the exposure variables $x_1, x_2, \ldots, x_g$ in Equation (14) are available from the individual rows of the database view. Upon multiplying the probabilities contributed by each row, we obtain the likelihood function $L_{W[i][j]}$ defined by:

$$L_{W[i][j]}(\alpha, \beta_1, \beta_2, ..., \beta_g) \tag{15}$$

Next, we estimate the values of the statistical parameters $\alpha, \beta_1, \beta_2, \ldots, \beta_g$ that maximize $L_{W[i][j]}$ using maximum likelihood estimation. We apply the unconditional likelihood technique instead of the conditional technique because the number of statistical parameters in the model is usually small relative to the number of rows in the database view. Furthermore, the conditional likelihood technique does not allow the estimation of the intercept term $\alpha$, which, as can be seen from Equations (12) and (13), is indispensable to estimating the probability that $W[i][j]$ takes values of 1 and 0, respectively. If we denote the parameters of the likelihood function $L_{W[i][j]}$ as $\theta = (\theta_1, \theta_2, \ldots, \theta_{g+1})$, then the values of the statistical parameters that maximize $L_{W[i][j]}(\theta)$ are the solutions of a system of equations of the form given by Equation (11).

In this case, $\theta_r$ is the $r^{\text{th}}$ individual parameter for $r \in \{1, 2, \ldots, g+1\}$. The solutions of the system of equations give the estimates of the statistical parameters $\alpha, \beta_1, \beta_2, ..., \beta_g$. With the statistical parameter estimates in hand, we use Equations (12) and (13) to estimate the probability that $W[i][j]$ takes the values 1 and 0, respectively, given the current values of the elements of matrix $W$. This is also an integral component of the estimation-inspection algorithm.

Estimating $p_1 = P(\varphi(W[i][j]) = 1 \mid W)$ and $p_0 = P(\varphi(W[i][j]) = 0 \mid W)$, and storing $p_1$ and $p_0$ in $V_{W[i][j]}[1]$ and $V_{W[i][j]}[0]$, respectively, fills both the positions of the stochastic vector $V_{W[i][j]}$. Iterating over every possible tuple of values of the exposure variables $x_1, x_2, \ldots, x_g$ associates each of them with a stochastic vector $V_{W[i][j]}$, which leads to the computation of the probability mass function $\Gamma_{W[i][j]}$.

**Algorithm 1** : Assess the normalcy of a network packet payload.

**Part I**

1: **for all** $W[i][j]$ that models a program variable that is defined **do**
2: **if** $W[i][j]$ models a continuous variable **then**
3: estimate the associated statistical parameters $\alpha_1$, $\alpha_2$, ..., $\alpha_h$, $\beta_1$, $\beta_2$, ..., $\beta_g$ using ordinal logistic regression and maximum likelihood estimation
4: **end if**
5: **if** $W[i][j]$ models a logical variable **then**
6: estimate the associated statistical parameters $\alpha$, $\beta_1$, $\beta_2$, ..., $\beta_g$ using dichotomous logistic regression and maximum likelihood estimation
7: **end if**
8: **end for**

**Part II**

1: $U \Leftarrow payload$
2: $Norm \Leftarrow true$
3: **for all** $W[i][j]$ such that $\varphi(W[i][j]) \in U$ **do**
4: $k \Leftarrow \varphi(W[i][j]) - min(W[i][j])$
5: **if** $W[i][j]$ models a continuous variable **then**
6: $p_k \Leftarrow \frac{1}{1+e^{-(\alpha_k + \sum_{a=1}^{g} \beta_a x_a)}} - \frac{1}{1+e^{-(\alpha_{k+1} + \sum_{a=1}^{g} \beta_a x_a)}}$
7: **end if**
8: **if** $W[i][j]$ models a logical variable and $k = 1$ **then**
9: $p_k \Leftarrow \frac{1}{1+e^{-(\alpha + \sum_{a=1}^{g} \beta_a x_a)}}$
10: **else if** $W[i][j]$ models a logical variable and $k = 0$ **then**
11: $p_k \Leftarrow 1 - \frac{1}{1+e^{-(\alpha + \sum_{a=1}^{g} \beta_a x_a)}}$
12: **end if**
13: **if** $p_k = 0$ **then**
14: $Norm \Leftarrow false$
15: break $for$ loop
16: **end if**
17: **end for**
18: **return** $Norm$

## 3.3 Estimation-Inspection Algorithm

The first part of the estimation-inspection algorithm (see Part I of Algorithm 1) is concerned with estimating the statistical parameters (intercept terms $\alpha$(s) and coefficient terms $\beta$(s)) and is, therefore, conducted during the learning phase. As indicated in Line 1 (Part I), the algorithm estimates a specific set of statistical parameters for each element of the matrix $W$ that models a program variable defined in a control system. As discussed above, the algorithm applies ordinal logistic regression integrated with maximum likelihood estimation on a

learning data set to estimate the intercept and coefficient terms of the ordinal logistic model developed for an element of matrix $W$ that models a continuous variable (Lines 2–4). The algorithm applies dichotomous logistic regression integrated with maximum likelihood estimation on a learning data set to estimate the intercept term and the coefficient terms of the dichotomous logistic model developed for an element of matrix $W$ that models a logical variable (Lines 5–7).

Part II of the algorithm is concerned with scrutinizing network packets in a process control network. To assess the normalcy of a network packet, the algorithm conducts its statistical analysis in relation to each variable that is written by the network packet (Line 3). The algorithm checks if the program variable written by the network packet is a continuous variable (Line 4) or a logical variable (Lines 8, 10). This information along with the value of index $k$ computed in Line 4 are used to identify: (i) the type of logistic model and, thus, the corresponding logistic regression formula applicable to the network packet; and (ii) the intercept terms $\alpha$(s) and coefficient terms $\beta$(s) of the applicable logistic model defined for the variable by the packet.

If the program variable written by the network packet is a continuous variable, the algorithm plugs the intercept terms $\alpha$(s) and coefficient terms $\beta$(s) along with the current values of the exposure variables $x_1, x_2, \ldots, x_g$ into the formula of the ordinal logistic model and produces an estimate of the normalcy probability of the specific value that the network packet writes to the continuous variable in question (Line 6). If the program variable written is a logical variable, the algorithm plugs the intercept term $\alpha$ and coefficient terms $\beta$(s) along with the current values of the exposure variables $x_1, x_2, \ldots, x_g$ into the formula of the dichotomous logistic model to estimate the normalcy probability of value 1 (Line 9) or value 0 (Line 11) depending on whether 1 or 0 is written to the logical variable, respectively.

If the normalcy probability of the value written to the program variable under consideration is greater than zero, the algorithm conducts its statistical analysis in relation to the next variable that the network packet under inspection will write, if any. If this is not the case, i.e., the estimate of the normalcy probability is equal to zero, the algorithm interrupts the scrutinization process and classifies the network packet as abnormal (Lines 13–16).

## 4. Experimental Evaluation

A small testbed was used to generate a data set for the inductive machine learning process used by the estimation-inspection algorithm and to conduct an experimental evaluation of the algorithm. The control system employed in the testbed comprised Linux PC-based PLCs [20], specifically, MatPLCs [22] installed on general-purpose Linux machines with x86 CPUs. Custom MatPLC modules were employed in the master mode to control and monitor a limited number of simulated components of an ABWR. These modules implemented control logic for processing MatPLC points (inputs, outputs, internal coils and registers). The MatPLC points were mapped to physical I/O parameters and,

therefore, represented the link between the MatPLC modules in master mode and the parameters of simulated ABWR components. Network communications were implemented using the Modbus protocol over TCP/IP.

Sensors and actuators were emulated using custom MatPLC modules running in the slave mode. A MatPLC human-machine interface (HMI) GNU image manipulation program toolkit (GTK) module was used to read and write MatPLC points corresponding to supervisory network operations of a power plant. We conducted continuous simulations [2] of the mechanisms used to insert or withdraw a control rod (namely, the joint operation of an AC induction motor that produces a torque and a ball screw that transforms rotational motion into linear motion), a motor-driven water pump used to inject water within the reactor core, and limited portions of the nuclear fission process that involve reactivity [21] and core flow (i.e., water in the reactor core).

A prototype implementation of the estimation-inspection algorithm was deployed and activated in the MatPLCs and the simulated ABWR components were run normally using the control system and network. The main purpose of the test was to assess if the algorithm would mistakenly classify normal network packets as abnormal and, thus, generate false positives. To assess the effectiveness of the algorithm in detecting attacks, a series of memory errors were inserted in the Modbus implementation running on the MatPLCs and attack code was developed to exploit the errors.

The attacks launched on the MatPLCs included stack overflow exploits with shellcode injection, stack overflow exploits with arc injection, heap overflow exploits with shellcode injection, frame pointer overwrites with shellcode injection, format bug exploits with shellcode injection that corrupted function pointers in the global offset table, indirect pointer overwrites with shellcode injection that corrupted function pointers in the global offset table, and exploits of out-of-boundary array indices with shellcode injection. Inertial attacks [10] were also mounted on the simulated AC induction motor along with exclusion attacks that violated a functional dependency between the (limited) simulated control rod insertion and withdrawal system and the (limited) simulated reactor feedwater system.

We obtained a false alarm rate of zero false positives/hour, which we believe is a clear indication of the need to test the estimation-inspection algorithm on a data set comprising network packets sniffed from the process control network of a real power plant. Conversely, this initial result may indicate that the algorithm has potential to be highly effective.

We also obtained a detection probability of 98%, i.e., 98% of the malicious network packets were detected by the algorithm. When possible, we crafted network packets so as to inject shellcode one byte at a time. A few of these bytes managed to pass undetected because they were indeed normal process data in defined states of the simulated ABWR components. All the network packets that injected memory addresses were detected by the algorithm. In summary, all the attacks launched in the test were detected by the estimation-inspection algorithm.

## 5. Conclusions

The estimation-inspection algorithm is intended to protect cyber-physical systems such as power plants from application-level computer network attacks. The algorithm uses statistical techniques to determine if the payload of a network packet that is about to be processed by a control system is normal or abnormal based on the evolution of the variable that the network packet will modify. Experimental results with a small testbed demonstrate that the algorithm yields a detection probability of 98% with a zero false positive rate.

It is necessary to conduct additional tests of the estimation-inspection algorithm. In particular, the algorithm should be tested using packets collected from the process control network of a real power plant.

## Acknowledgements

This work was supported in part by NSF Grant CNS 0614771. The research of Julian Rrushi was partially supported by scholarships from the University of Milan and $(ISC)^2$.

## References

[1] A. Cardenas, S. Amin and S. Sastry, Research challenges for the security of control systems, *Proceedings of the Third USENIX Workshop on Hot Topics in Security*, 2008.

[2] F. Cellier and E. Kofman, *Continuous System Simulation*, Springer, New York, 2006.

[3] S. Cheung, B. Dutertre, M. Fong, U. Lindqvist, K. Skinner and A. Valdes, Using model-based intrusion detection for SCADA networks, *Proceedings of the SCADA Security Scientific Symposium*, 2007.

[4] General Electric Company, Advanced Boiling Water Reactor (ABWR), Fairfield, Connecticut (www.gepower.com/prod_serv/products/nuclear_energy/en/new_reactors/abwr.htm).

[5] D. Hosmer and S. Lemeshow, *Applied Logistic Regression*, Wiley, Hoboken, New Jersey, 2000.

[6] International Electrotechnical Commission, IEC 61850-7-410: Communication Networks and Systems for Power Utility Automation, Part 7-410: Hydroelectric Power Plants – Communication for Monitoring and Control, Geneva, Switzerland, 2007.

[7] H. Javitz and A. Valdes, The NIDES Statistical Component Description and Justification, SRI Project 3131 Annual Report, SRI, Menlo Park, California, 1994.

[8] S. Kay, *Fundamentals of Statistical Signal Processing, Volume 2: Detection Theory*, Prentice Hall, Upper Saddle River, New Jersey, 1998.

[9] D. Kleinbaum, L. Kupper, L. Muller and A. Nizam, *Applied Regression Analysis and Multivariable Methods*, Duxbury Press, Pacific Grove, California, 2007.

[10] J. Larsen, SCADA security, presented at *Blackhat DC*, 2008.

[11] E. Lehmann and G. Casella, *Theory of Point Estimation*, Springer, New York, 2003.

[12] J. Marcum, A statistical theory of target detection by pulsed radar, *IRE Transactions on Information Theory*, vol. 6(2), pp. 59–267, 1960.

[13] Modbus IDA, MODBUS Application Protocol Specification v1.1a, North Grafton, Massachusetts (www.modbus.org/specs.php), 2004.

[14] M. Naedele and O. Biderbost, Human-assisted intrusion detection for process control systems, *Proceedings of the Second International Conference on Applied Cryptography and Network Security*, pp. 216–225, 2004.

[15] E. Naess, D. Frincke, A. McKinnon and D. Bakken, Configurable middleware-level intrusion detection for embedded systems, *Proceedings of the Second International Workshop on Security in Distributed Computing Systems*, vol. 2, pp. 144–151, 2005.

[16] D. Nicol and P. Heidelberger, Parallel execution for serial simulators, *ACM Transactions on Modeling and Computer Simulation*, vol. 6(3), pp. 210–242, 1996.

[17] J. Rrushi and R. Campbell, An intrusion detection system for operation in nuclear power plants, presented at the *Fourth ITI Workshop on Dependability and Security*, 2007.

[18] J. Rrushi and R. Campbell, Using deception to facilitate intrusion detection in nuclear power plants, *Proceedings of the Third International Conference on Information Warfare and Security*, 2008.

[19] J. Rrushi and K. Kang, Mirage theory: A deception approach to intrusion detection in process control networks, *Proceedings of the NATO Symposium on Information Assurance for Emerging and Future Military Systems*, 2008.

[20] K. Stouffer, J. Falco and K. Scarfone, Guide to Industrial Control Systems (ICS) Security, Special Publication 800-82, Final Public Draft, National Institute of Standards and Technology, Gaithersburg, Maryland, 2008.

[21] U.S. Department of Energy, DoE Fundamentals: Handbook of Nuclear Physics and Reactor Theory, DOE-HDBK-1019/1-93, Washington, DC, 1993.

[22] C. Wuollet, A. Romanenko, H. Jack, J. Baum, J. Orozco and M. de Sousa, MatPLC (mat.sourceforge.net), 2006.