# 一种基于状态迁移图的工业控制系统异常检测方法

吕雪峰 [1, 2] 谢耀滨 [2]

**摘 要** 基于状态的工业控制系统入侵检测方法以其高准确率受到研究者的青睐, 但是这种方法往往依赖专家经验事先定义系统的临界状态, 且处理不了系统状态变量较多的情况. 针对这一问题, 提出一种新的基于状态迁移图的异常检测方法. 该方法利用相邻数据向量间的余弦相似度和欧氏距离建立系统正常状态迁移模型, 不需要事先定义系统的临界状态, 并通过以下两个条件来判定系统是否处于异常: 1) 新的数据向量对应的状态是否位于状态迁移图内; 2) 前一状态到当前状态是否可达. 文章建立了恶意数据攻击模型, 并以田纳西–伊斯曼 (Tennessee-eastman, TE) 过程 MATLAB 模型作为仿真平台进行了仿真测试. 仿真结果表明, 该方法即使在系统遭受轻微攻击的情况下也有较好的检测结果, 且消耗较少的时空资源.

**关键词** 工业控制系统, 状态迁移图, 异常检测, 田纳西–伊斯曼过程

**引用格式** 吕雪峰, 谢耀滨. 一种基于状态迁移图的工业控制系统异常检测方法. 自动化学报, 2018, **44**(9): 1662−1671

**DOI** 10.16383/j.aas.2017.c160832

## An Anomaly Detection Method for Industrial Control Systems via State Transition Graph

LV Xue-Feng[1, 2] XIE Yao-Bin[2]

**Abstract** State-based intrusion detection method for industrial control system is favored owing to its high accuracy, but this kind of method often relies on some critical states defined by expert experience beforehand and cannot deal with systems containing a number of variables. To handle this problem, a new anomaly detection method based on state transition graph is proposed. The proposed method constructs a normal state transition model of the system depending on the cosine similarity and Euclidian distance between two adjacent data vectors without any predefined critical states, and can determine whether the system is in the normal state or not according to the following two conditions: 1) whether or not the current state calculated by the new data vector is in the state transition graph; 2) whether or not the previous state can reach the current state. To evaluate the method, a false data injection model is established and tested on a Tennessee-Eastman (TE) process simulated by MATLAB. The result shows that even when the attack is insensitive the method can still get good detection result and consume little time and space resource.

**Key words** Industrial control system, state transition graph, anomaly detection, Tennessee-Eastman (TE) process

**Citation** Lv Xue-Feng, Xie Yao-Bin. An anomaly detection method for industrial control systems via state transition graph. *Acta Automatica Sinica*, 2018, **44**(9): 1662−1671

工业控制系统 (Industrial control system) 广泛应用于国家基础设施, 工控系统一旦遭到破坏可能会造成难以估量的经济损失甚至人员伤亡. 一个典型的工业控制网路可分为企业网络层、控制网络层和现场网络层, 如图 1 所示.

近年来, 由于 IT 系统的软件和硬件技术不断集成到工控领域, IT 领域的一些漏洞和后门也出现在工控系统上. 以智能制造为核心的工业 4.0, 在推动工业转型的同时也使得工控系统面临更多来自互联网的威胁[1]. 2010 年, 世界首个网络超级破坏武器 "震网" 病毒[2] 被检测出来, 证明了工控系统是可被攻击并被利用的. 此后又相继爆发了 "毒区"、"火焰" 等病毒和一连串的工控系统入侵事件. 工控信息安全受到越来越多的重视, 针对工控系统入侵检测的研究也成为目前的一个热点.

由于工控保护机制的存在, 大的破坏性攻击容易被检测出来, 因此对入侵检测的研究越来越集中于隐蔽攻击[3−5]. 当前, 工控系统入侵检测的研究方法主要包含三类: 1) 基于概率统计的方法[6−7]; 2) 基于机器学习的方法[8−12]; 3) 基于状态的方法[13−16].

基于状态的方法因为更好地考虑了工控系统的物理特性, 检测准确度高, 目前大部分研究都是基于此类方法[13]. 基于状态的方法通过历史数据和专家经验定义系统的临界状态, 实时监控系统当前状态与临界状态的距离以判断系统是否处于危险之中.

收稿日期 2016-12-22 录用日期 2017-05-22
Manuscript received December 22, 2016; accepted May 22, 2017
本文责任编委 胡昌华
Recommended by Associate Editor HU Chang-Hua
1. 数学工程与先进计算国家重点实验室 郑州 450001 2. 解放军信息工程大学 郑州 450001
1. State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450001 2. PLA Information Engineering University, Zhengzhou 450001

Fovino 等[15] 通过监控系统状态的变化来检测复杂攻击, 创立了系统的虚拟镜像作为系统的内部表示, 并用规则语言描述系统的临界状态, 但是仅考虑了离散输入输出数据, 没有考虑连续数据. Carcano 等[17] 进一步考虑了连续输入和输出, 但在输入输出数据量庞大的情况下, 很难定义系统的临界状态. 针对这一问题, Khalili 等[13] 提出了一个系统化的解决方案 SysDetect, 该方案基于 Apriori 算法, 通过专家经验的判定可显著减少算法下一次迭代产生的候选临界状态数. 但是该方法依然不能很好地处理高维数据, 且在一定程度上依赖人工判定.

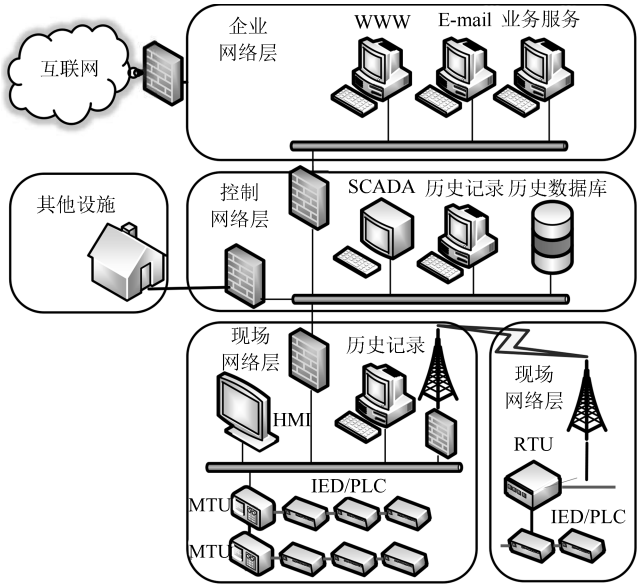



图 1　典型的工业控制系统网络架构

Fig. 1　Typical network architecture for industrial control system

针对当前工控入侵检测系统存在的问题, 本文提出一种新的基于状态迁移图的异常检测方法. 利用工控系统与物理世界交互的特性, 采用数据驱动的方法进行建模, 即利用系统运行时系统变量的数据来建立检测模型. 但考虑到数据维度可能过高, 不直接以系统变量数据来描述系统运行状态, 而是以相邻数据向量间的余弦相似度和欧氏距离来表征, 因而可以处理高维数据. 状态迁移图刻画系统运行过程中的正常模型, 根据正常历史数据样本训练得出, 所以不需要依赖专家预先定义系统的临界状态.

## 1　恶意数据攻击建模

一个工业控制过程可简单地用图 2 来表示. 其中传感器和执行器易成为攻击者的攻击目标, 因为它们直接与物理过程相连, 一旦遭到攻击将会产生不可估量的后果, 例如 “震网” 病毒恶意篡改了伊朗核工厂离心机的转速, 使核工厂被迫停工.

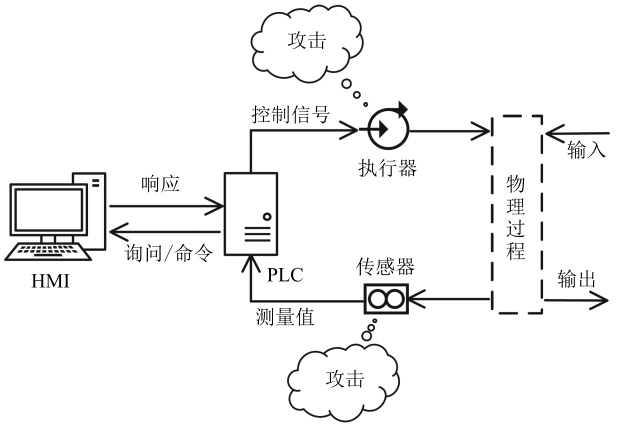



图 2　工业控制过程: 传感器和执行器易成为攻击目标

Fig. 2　Industrial control process: sensors and actuators are vulnerable targets

由于工控系统的物理特性, 恶意数据注入成为一种简单且收益高的方式. 攻击者通过攻击执行器和传感器, 注入恶意数据, 修改变量的值, 对系统造成破坏.

系统变量分为传感变量和操纵变量, 传感变量的值由传感器测量取得, 而操纵变量则由控制器通过系统参数和传感变量值计算得出. 记系统含 $l$ 个测量变量, 用集合 $\boldsymbol{y}=\{y_k|k=1,2,\cdots,l\}$ 表示, 含 $p$ 个操纵变量, 用集合 $\boldsymbol{u}=\{u_k|k=1,2,\cdots,p\}$ 表示. 则系统变量集合可用 $\boldsymbol{s}=\{s_j|j=1,2,\cdots,l+p\}$ 表示. 根据攻击者试图破坏系统速度的快慢, 将恶意数据注入攻击分为快速注入和慢速注入.

快速注入使得系统变量在短时间内发生较大改变, 以最快速度造成破坏, 表现为最大/最小值攻击.

$$\tilde{s}_i(t)=\begin{cases}s_i(t), & t\notin T_a\\ {s_i}^{\max}, & t\in T_a\end{cases} \tag{1}$$

$$\tilde{s}_i(t)=\begin{cases}s_i(t), & t\notin T_a\\ {s_i}^{\min}, & t\in T_a\end{cases} \tag{2}$$

其中, $T_a$ 为攻击时段.

慢速注入缓慢改变系统变量的值, 以达到潜伏的目的, 可以表现为偏置注入和几何注入. 偏置注入连续注入多个较小的常量.

$$\tilde{s}_i(t)=\begin{cases}s_i(t)+c_i, & t\in T_a\\ s_i(t), & t\notin T_a\end{cases} \tag{3}$$

其中, $c_i$ 表示常量, 通常取值较小. 几何注入逐渐增大系统变量的改变幅度, 具有一定的潜伏性, 同时不失破坏性, 可表现为指数形式的注入.

$$\tilde{s}_i(t)=\begin{cases}s_i(t)+ab^{t-t_0}, & t\in T_a\\ s_i(t), & t\notin T_a\end{cases}\tag{4}$$

其中, $a$, $b$ 可以是常数, 也可以是变量, $t_0$ 为常数.

以上只是列举了三种可能的攻击形式, 实际上恶意数据注入攻击可以表现为更多的形式.

## 2 基于状态迁移图的异常检测方法

由于工控系统的诸多限制, 例如有限的内存、有限的计算能力、较高的实时性要求等, 工控入侵检测系统必须是轻量级的, 检测规则或检测模型应设置得相对简单. 本文采用一个状态迁移图的检测模型, 状态迁移图的状态数与数据向量的维度无关, 因而可以处理高维的数据向量.

### 2.1 状态表示

实际的工控系统中, 状态变量可能有很多个, 因此数据向量的维度可能很高, 而现有的基于状态的入侵检测算法无法很好地处理数据维度较高的情况. 考虑控制系统的运行模式, 控制器根据前一时刻传感变量的值, 计算出操纵变量的值, 操纵变量的值再作用于物理系统使之发生变化, 传感变量随着系统的运动而变化, 在下一个采样周期将传感变量的值传给控制器, 如此循环往复. 通过分析这个过程可以发现, 系统变量在当前时刻的取值很大程度上取决于上一时刻的取值, 因此正常情况下, 相邻两个数据向量间的变化不会太大.

相邻数据向量之间的变化反应了系统变量运行的时间特性, 本文以余弦相似度和欧氏距离来衡量这种变化. 余弦相似度反应数据向量方向的变化, 常用于文档聚类和信息提取. 欧氏距离衡量两个数据向量绝对距离的远近.

选择作为系统状态表征的特征应该是相互关联的, 即相邻数据向量间的余弦相似度和欧氏距离应该是相互关联的. 这样当余弦相似度在某个区间取值时, 欧氏距离只位于某些特定区间, 而不会在所有区间取值. 如图 3 所示, 假设余弦相似度取值区间为 $A$, $B$, $C$, 欧氏距离取值区间为 $D$, $E$, $F$, 二者在二维空间所有可能的区域为 $1\sim9$, 实际取值区域为 1, 4, 5, 9. 当余弦相似度取值区间为 $B$ 时, 欧氏距离只在区间 $D, E$ 取值, 而不会在 $F$ 区间取值, 若欧氏距离在 $F$ 区间取值就可以认为发生了异常. 这样可以更加细致地刻画数据的轮廓, 而不是将所有可能的区域为 $1\sim9$ 都视为可接受的区域. 余弦相似度和欧氏距离的关联性如图 4 所示, 其中余弦相似度和欧氏距离采用 Normal 数据集计算得出 (经过归一化处理), 图 4 展现了前 30 组数据. 从图 4 可以看出, 余弦相似度和欧氏距离的变化基本相反, 当余弦相似度在高位取值时, 欧氏距离总在低位取值, 证明了系统变量数据向量间的余弦相似度和欧氏距离是相互关联的.

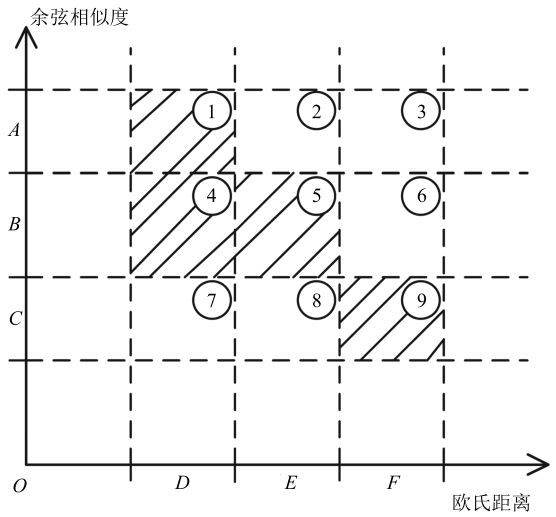



图 3 余弦相似度和欧氏距离取值示意

Fig. 3 Possible taking-value for cosine similarity and Euclidian distance

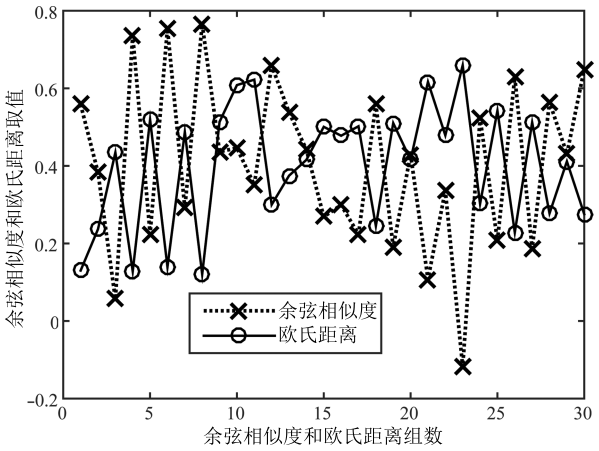



图 4 余弦相似度和欧氏距离关联性示意

Fig. 4 Relevance between cosine similarity and Euclidian distance

本文以数据向量间的余弦相似度和欧氏距离组成的二元组来表征系统运行状态, 避免了直接对数据进行建模, 将高维的数据向量转化为二维数据, 有效降低了计算复杂度. 同时, 考虑数据的时间特性, 更加符合工控系统的动态特性.

### 2.2 状态矩阵求解

为了求解出状态迁移图, 需要知道图的顶点和各顶点之间的转换关系. 状态迁移图的顶点表示了系统可能的状态, 本文用一个状态矩阵求解状态迁移图, 需要知道图的顶点和各顶点之间的转换关系.

状态迁移图的顶点表示了系统可能的状态, 本文以状态矩阵 $STATE$ 表示可能的状态, $STATE$ 的每一行称为一个状态点, 一个状态点表示一个状态, 对应状态迁移图中的一个顶点.

#### 2.2.1　余弦相似度和欧氏距离的求解

在某一时刻所有系统变量取值的集合称为一样本点, 例如 $\boldsymbol{m}_i = \{s_{1,i}, s_{2,i}, \cdots, s_{l+p,i}\}$ 表示系统在 $i$ 时刻的样本点, $s_{k,t} = s_k(t)$ 为第 $k$ 个系统变量在 $t$ 时刻的取值. 为了计算方便, 用行向量的形式来表示样本点, 即 $\boldsymbol{m}_i = [s_{1,i}, s_{2,i}, \cdots, s_{l+p,i}]$. 假设训练样本点数为 $n$, 相邻样本点 (经过 z-score 标准化的样本点) 间的余弦相似度由式 (5) 计算得出.

$$cossim_i = \frac{\boldsymbol{m}_i \times \boldsymbol{m}_{i+1}^{\mathrm{T}}}{\|\boldsymbol{m}_i\| \times \|\boldsymbol{m}_{i+1}\|}, \quad i = 1, 2, \cdots, n-1 \tag{5}$$

令

$$\boldsymbol{cossim} = [cossim_1,\ cossim_2,\ \cdots,\ cossim_{n-1}]^{\mathrm{T}}$$

$\boldsymbol{cossim}$ 是所有训练样本点任意相邻两点间余弦相似度组成的列向量. 相邻样本点间的欧氏距离由式 (6) 计算得出.

$$ed_i = \sqrt{\boldsymbol{m}_i \times \boldsymbol{m}_{i+1}^{\mathrm{T}}}, \quad i = 1, 2, \cdots, n-1 \tag{6}$$

令 $\boldsymbol{ed} = [ed_1,\ ed_2,\ \cdots,\ ed_{n-1}]^{\mathrm{T}}$, $\boldsymbol{ed}$ 是所有训练样本点任意相邻两点间欧氏距离组成的列向量.

#### 2.2.2　$\boldsymbol{cossim}$ 和 $\boldsymbol{ed}$ 量化

由于 $\boldsymbol{cossim}$ 和 $\boldsymbol{ed}$ 中的元素都是实数, 这样通过二者确定的状态图中的状态数目将是无限个, 所以需要对它们进行量化处理, 使得 $\boldsymbol{cossim}$ 和 $\boldsymbol{ed}$ 中的元素只能取一些固定值. 可以用一些量化阈值将 $\boldsymbol{cossim}$ 和 $\boldsymbol{ed}$ 的取值分成数个量化区间, 同一量化区间的点用相同的值来替换. 本文用 $intervals$ 表示 $\boldsymbol{cossim}$ 和 $\boldsymbol{ed}$ 量化区间的数量.

为便于处理, 首先对 $\boldsymbol{cossim}$ 和 $\boldsymbol{ed}$ 进行归一化处理. 将 $\boldsymbol{cossim}$ 归一化到 $[-1, 1]$ 区间, 得到 $\boldsymbol{comsim}'$; $\boldsymbol{ed}$ 归一化到 $[0, 1]$ 区间, 得到 $\boldsymbol{ed}'$. 令 $\boldsymbol{comsim}'$ 的量化阈值集为

$$\boldsymbol{r} = \{r_0, r_1, r_2, \cdots, r_{\mathrm{intervals}}\} \tag{7}$$

$\boldsymbol{ed}'$ 的量化阈值集为

$$\boldsymbol{d} = \{d_0,\ d_1, d_2, \cdots, d_{\mathrm{intervals}}\} \tag{8}$$

按照各量化区间点数相等的原则计算 $\boldsymbol{r}$ 和 $\boldsymbol{d}$. 定义函数 $k : \mathbf{R} \to \mathbf{R}$, $k(P)$ 表示某实数集合 $P$ 中元素的个数. 定义函数

$$f(x) = k(\{j | j \le x, j \in \boldsymbol{cossim}'\})$$

$$g(x) = k(\{j | j \le x, j \in \boldsymbol{ed}'\})$$

$\boldsymbol{r}$ 和 $\boldsymbol{d}$ 的计算如下:

$$r_k = \begin{cases} -1, & k = 0 \\ \arg\min\limits_{x \in [-1,1]} f(x) - f(r_{k-1}) = \dfrac{n}{intervals}, & k = 1, 2, \cdots, intervals - 1 \\ 1, & k = intervals \end{cases} \tag{9}$$

$$d_k = \begin{cases} 0, & k = 0 \\ \arg\min\limits_{x \in [0,1]} g(x) - g(d_{k-1}) = \dfrac{n}{intervals}, & k = 1, 2, \cdots, intervals - 1 \\ 1, & k = intervals \end{cases} \tag{10}$$

#### 2.2.3　状态矩阵生成

确定量化阈值后, 根据量化阈值对 $\boldsymbol{comsim}'$ 和 $\boldsymbol{ed}'$ 进行量化处理.

$$qcossim_i = \arg\min_{r \in \boldsymbol{r}} cossim_i \le r, \quad i = 1, 2, \cdots, n-1 \tag{11}$$

$\boldsymbol{qcomsim}' = [qcossim_1 | qcossim_2 | \cdots | qcossim_{n-1}]^{\mathrm{T}}$ 为量化后的 $\boldsymbol{comsim}'$.

$$qed_i = \arg\min_{d \in \boldsymbol{d}} ed_i \le d, \quad i = 1, 2, \cdots, n-1 \tag{12}$$

$\boldsymbol{qed}' = [qed_1 | qed_2 | \cdots | qed_{n-1}]$ 为量化后的 $\boldsymbol{ed}'$. 最终的状态矩阵为 $STATE = [\boldsymbol{qcossim}' | \boldsymbol{qed}']$.

### 2.3　状态迁移图生成

状态迁移图是一个有向图, 作为异常检测模型, 其顶点由 $STATE$ 确定, 边用邻接矩阵表示.

#### 2.3.1　顶点确定

由于对相邻样本点间的余弦相似度和欧氏距离进行了量化, 状态矩阵 $STATE$ 中含有相同的状态点. 相同的状态点对应状态迁移图中的同一个顶点, 应将其合并, 具体做法为:

**步骤 1.** 用自然数对 $STATE$ 中的每个状态点进行标号, 称为状态号, 并用向量 $\boldsymbol{sign}$ 来表示, 例如 $\boldsymbol{sign}(k)$ 表示 $STATE$ 中的第 $k$ 个状态点对应的状态号. 状态号唯一标明了状态迁移图中的状态 (即顶点).

**步骤 2.** 令 $STATE$ 第 1 个状态点的状态号为 1, 即 $\boldsymbol{sign}(1) = 1$, 令状态数 $sn = 1$.

**步骤 3.** 从第 2 个状态点开始, 对第 $k$ 个状态点遍历 $STATE$ 中该状态点之前的状态点, 将该状

态点与之前的状态点进行比较, 若找到相同的状态点 $j$ 则停止遍历, 将该点的状态号设为第 $j$ 个状态点的状态号, 即 $\boldsymbol{sign}(k) = \boldsymbol{sign}(j)$; 若没有找到相同的状态点, 则令 $sn = sn + 1$, 将该点的状态号设为 $sn$, 即 $\boldsymbol{sign}(k) = sn$.

经过步骤 1 ~ 3 后, $STATE$ 中相同的状态点具有相同的状态号, 最终的状态数为 $sn$. 状态迁移图的顶点对应 $STATE$ 中状态号为 1 ~ $sn$ 的状态点.

#### 2.3.2 邻接矩阵确定

用 $NM$ 表示邻接矩阵, 邻接矩阵反映了状态迁移图中的状态转移关系, 若 $NM_{i,j} = 1$, 则状态迁移图中的第 $j$ 个顶点到第 $j$ 个顶点可达; 若 $NM_{i,j} = 0$, 则第 $i$ 个顶点到第 $i$ 个顶点不可达. 由于样本点是随时间增长逐渐采样而来的, 因此前一样本点到当前样本点总是可达的. 相应地, $STATE$ 中前一状态点到当前状态点也总是可达的. 根据这一原则可以求解出邻接矩阵 $NM$, 具体做法为:

**步骤 1.** 将 $NM$ 置为全零.

**步骤 2.** 根据前一样本点 (状态点) 到当前样本点 (状态点) 可达的原则, 令 $NM(\boldsymbol{sign}(i), \boldsymbol{sign}(i+1)) = 1$, 由此计算出 $NM$.

确定了顶点和邻接矩阵后, 就能生成状态迁移图.

### 2.4 基于状态迁移图的在线检测

状态迁移图生成后, 可以据此进行在线异常检测. 基于状态迁移图的在线检测流程如图所示, 具体步骤如下:

**步骤 1.** 获取当前样本点.

**步骤 2.** 计算当前样本点与上一样本的余弦相似度 $t\_cossim_i$ 和欧氏距离 $t\_ed_i$.

**步骤 3.** 按照训练样本的标准对 $t\_cossim_i$ 和 $t\_ed_i$ 进行归一化处理, 得到 $qt\_cossim_i$ 和 $qt\_ed_i$.

**步骤 4.** 遍历 $STATE$, 判断 $STATE$ 中是否有和当前点相同的点. 若存在, 根据 $\boldsymbol{sign}$ 确定当前状态点对应的状态号.

**步骤 5.** 若当前状态点对应的状态不在状态迁移图内, 产生告警, 记录异常类别为 "异常 1"; 若当前点对应的状态位于状态图内, 即 $STATE$ 中有和当前点相同的点, 则根据状态迁移图判断上一节点对应状态到当前节点对应状态是否可达. 若不可达, 则产生告警, 记录异常类别为 "异常 2"; 若可达, 继续下一个样本点的检测.

## 3 实验与测试

目前工控领域没有一个标准的测试集, 而实际工控系统在攻击下的数据样本很难获取. 针对这一情况, 本文采用田纳西–伊斯曼 (Tennessee-eastman, TE) 过程[18] 的 MATLAB 模型作为仿真平台. 根据第 1 节建立的恶意数据攻击模型, 选定斜坡注入和偏置注入两种攻击形式, 在特定时刻对 TE 过程实施攻击, 生成系统正常运行数据和攻击情形下的数据, 并用第 2.4 节描述的检测方法进行检测.

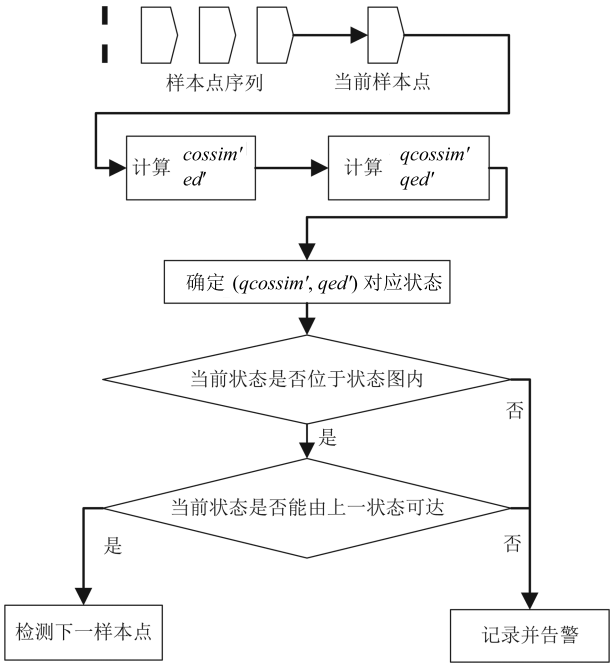



图 5 基于状态迁移图的在线检测流程

Fig. 5 Online detection process based on state transition graph

### 3.1 TE 仿真模型

TE 过程的原型由 Downs 和 Vogel 在 1993 年根据一个真实的化工过程创建. TE 过程主要部件有冷凝器、反应器、循环压缩机、汽提塔和气液分离器等. 该过程包含两个放热反应和两个副反应, 生成产品 $G$, $H$ 和副产品 $F$. 四种化学反应如下所示:

$$A\,(g) + C\,(g) + D\,(g) \rightarrow G\,(liq) \tag{13}$$

$$A\,(g) + C\,(g) + E\,(g) \rightarrow H\,(liq) \tag{14}$$

$$A\,(g) + E\,(g) \rightarrow F\,(liq) \tag{15}$$

$$3D\,(g) \rightarrow 2F\,(liq) \tag{16}$$

TE 过程包含 41 个测量变量和 12 个操纵变量, 是一个多变量, 多数据, 复杂的非线性控制系统, 常用于多变量控制、非线性控制和故障诊断等领域. TE 过程的 MATALB 模型由 Ricker[19] 给出, 该模型采用文献 [20] 中的控制策略, 仿真时间为 48 h, 采样时间为 3 min, 每次仿真共产生 960 组数据, 每组数据中都包含高斯白噪声. TE 过程的工艺流程如图 6 所示.

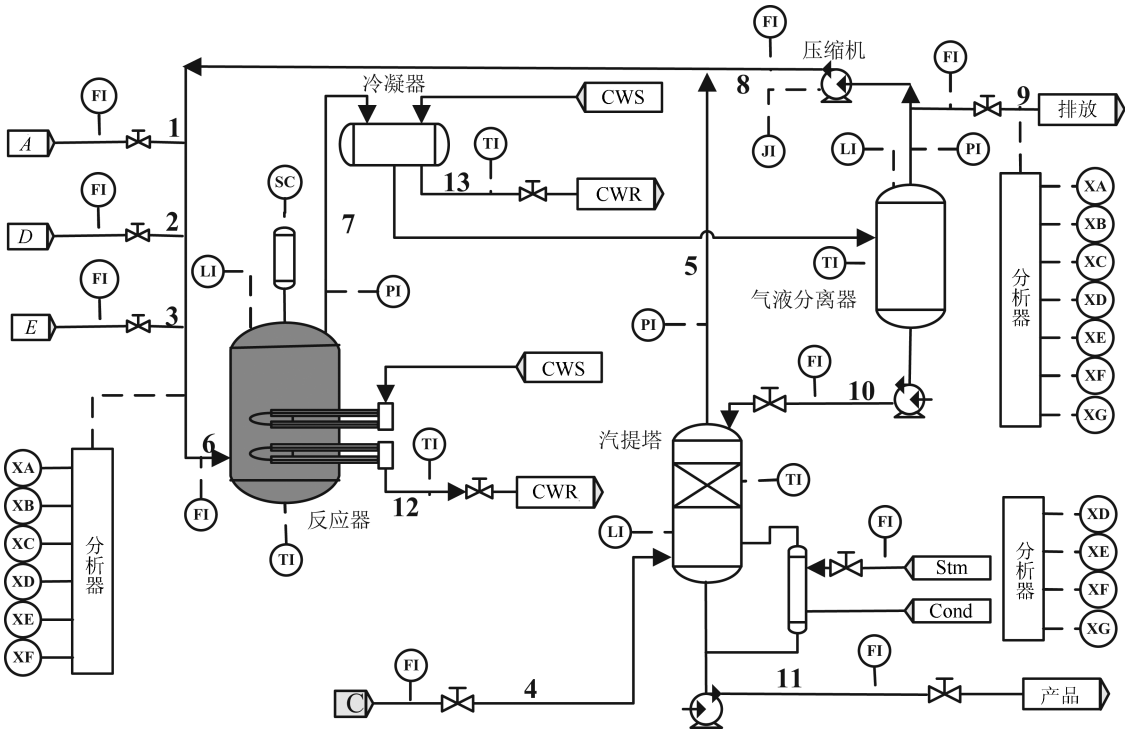



图 6　基于状态迁移图的在线检测流程

Fig. 6　Online detection process based on state transition graph

### 3.2　测试数据

反应器是 TE 过程的核心部件, 是发生化学反应的地方, 对温度的要求很高. 因此本文选定攻击对象为反应器的温度传感器, 并根据第 1 节所述攻击模型, 选定偏置注入和斜坡注入两种攻击方式.

$$\tilde{s}_i(t)=\begin{cases}s_i(t)+c, & t\in T_a\\ s_i(t), & t\notin T_a\end{cases} \tag{17}$$

$$\tilde{s}_i(t)=\begin{cases}s_i(t)+k(t-t_0), & t\in T_a\\ s_i(t), & t\notin T_a\end{cases} \tag{18}$$

其中, 偏置注入中 $c$ 为常数, 而斜坡注入在 $t_0$ 时注入斜率为 $k$ 的斜坡信号. 反应器温度控制回路如图 7 所示, 两个输入分别代表反应器温度初始设定值 (用输入点 9 表示) 和反应器温度 (用 xmeas 9 表示), 输出为反应器冷却水流量 (用 xmv10 表示). 该回路通过反应器温度测量值和反应器温度初始设定值计算冷却水流量, 以此来控制反应器温度. 为达到攻击反应器温度的目的, 在反应器温度输入端口增加 Add 模块, 将其与攻击信号叠加, 攻击信号通过延时模块在特定时刻发挥作用. 图 8 为加入攻击信号后的反应器温度控制回路, 延时模块时间设置为 20 h, 攻击信号为常数, 代表偏置注入信号. 攻击信号可以替换为其他信号, 以此设计更多的攻击形式.

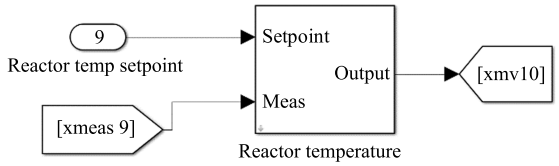



图 7　反应器温度控制回路

Fig. 7　Control loop for reactor temperature

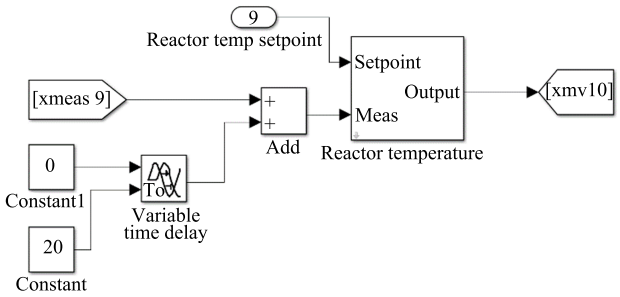



图 8　加入攻击信号的反应器温度控制回路

Fig. 8　Control loop for reactor temperature added with attack signal

采集系统正常运行时各变量的数据, 生成数据集合 Normal. 选定式 (17) 中 $c$ 为 0.01, 0.1 和 1, 在系统仿真 20 h 加入注入攻击, 生成偏置注入下的数据集 Dataset1, Dataset2, Dataset3. 选定式 (18) 中 $k$ 为 0.01, 0.1 和 1, 在仿真 20 h 实施斜坡攻击, 生成斜坡注入下的数据集 Dataset4, Dataset5,

Dataset6. 各数据集及其对应参数见表 1. 这些数据集将用来测试本文所提的方法.

表 1 测试数据集及其参数
Table 1 The test data set and the corresponding parameters

| 实验数据 | $c$ | $k$ |
|---|---|---|
| Normal | 0 | 0 |
| Dataset1 | 0.01 | 0 |
| Dataset2 | 0.1 | 0 |
| Dataset3 | 1 | 0 |
| Dataset4 | 0 | 0.01 |
| Dataset5 | 0 | 0.1 |
| Dataset6 | 0 | 1 |

为直观感受加入攻击信号后对反应器温度的影响, 用 MATLAB 采集生成了反应器温度在各种情况下的温度–时间变化曲线, 分别如图 9～11 所示. 图 9 为正常工况条件下的温度–时间变化曲线, 可以看出反应器温度稳定在 123 ℃左右. 图 10 为斜坡注入 $k = 0.01$ 情况下的温度变化情况, 可以看到在 20 h 后反应器温度呈线性下降趋势. 图 11 为偏置注入 $c = 0.1$ 情况下的温度–时间变化曲线, 反应器温度在 20 h 突然下降, 之后则稳定下来.

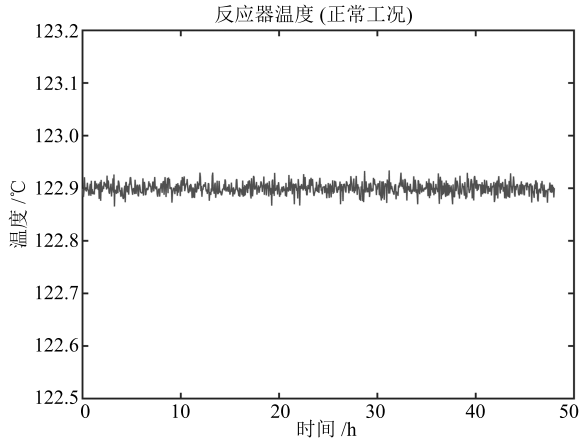



图 9 正常工况下反应器温度随时间变化情况
Fig. 9 Reactor temperature varies with time under normal condition

### 3.3 状态迁移图生成测试

由于状态迁移图通过正常数据训练得出, 所以用表 1 中的 Normal 数据集来进行训练. 根据第 2.2 节的分析, 生成的状态迁移图与余弦相似度和欧氏距离的量化区间数 $intervals$ 相关, 因此本文不断改变 $intervals$ 的取值, 进行了多组状态图测试. 首先

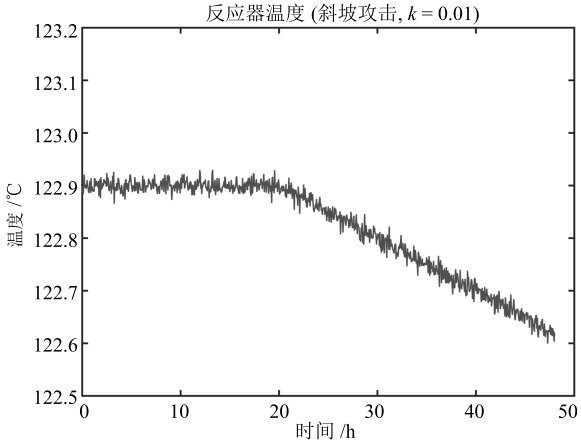



图 10 斜坡注入工况 ($k = 0.01$) 下, 反应器温度随时间变化情况
Fig. 10 Reactor temperature varies with time under ramp signal injection with $k$ set at 0.01

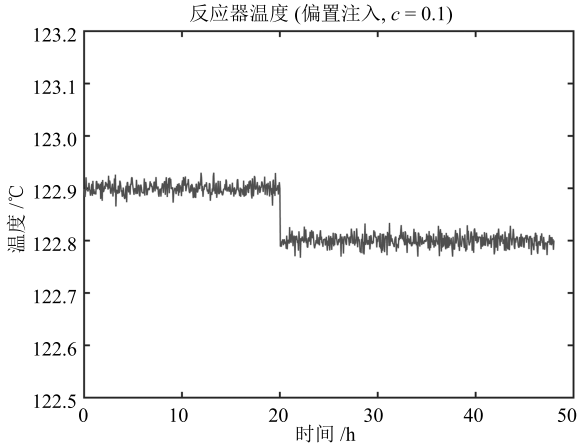



图 11 偏置注入工况 ($c = 0.1$) 下, 反应器温度随时间变化情况
Fig. 11 Reactor temperature varies with time under bias signal injection with $c$ set at 0.1

按照第 2.2 节和第 2.3 节的方式对 Normal 数据集进行处理, 然后设定量化区间数 $intervals$, 计算状态矩阵, 最后生成状态迁移图. 状态迁移图的顶点数和边数随 $intervals$ 的变化情况如表 2 所示.

从表 2 可以看出, 随着 $intervals$ 的增大, 状态迁移图的顶点数和边数也不断增大. 而 $intervals$ 越大, 意味着将余弦相似度和欧氏距离的量化区间划分得越小, 由于状态迁移图的检测由顶点和边来决定, 所以顶点数和边数越多, 意味着检测规则的粒度越细, 相应地对异常会更加敏感, 此外还会额外增加资源开销. 图 12 和图 13 分别显示了 $intervals$ 为 5 和 8 时用 MATLAB 生成的状态迁移图.

### 3.4 异常检测性能测试

按照第 2.4 节的在线检测步骤, 用表 1 的测试

表 2　状态迁移图的顶点数和边数随 *intervals* 的变化

Table 2　The nodes and triangles number of state transition graph varies with *intervals*

| *intervals* | 顶点数 | 边数 |
|---|---|---|
| 3 | 9 | 48 |
| 5 | 20 | 147 |
| 8 | 44 | 315 |
| 10 | 73 | 677 |
| 12 | 97 | 782 |
| 15 | 143 | 871 |

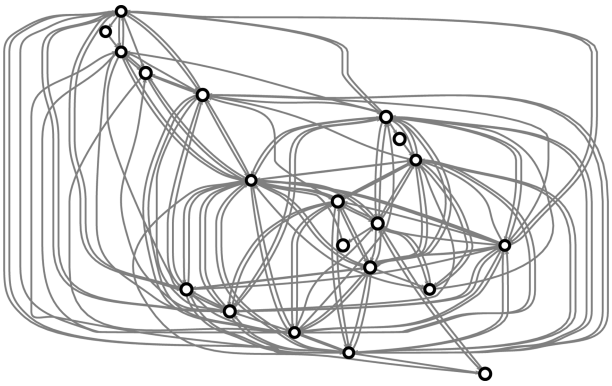


图 12　*intervals* 为 5 时的状态迁移图

Fig. 12　The state transition graph when *intervals* is equal to 5

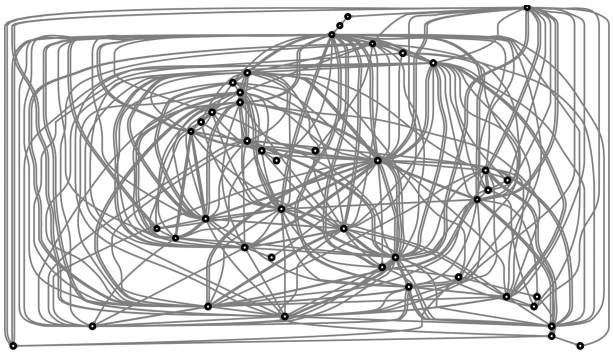


图 13　*intervals* 为 8 时的状态迁移图

Fig. 13　The state transition graph when *intervals* is equal to 8

数据对状态迁移图检测模型的检测效果进行测试, 选定 *intervals* 为 5, 8, 10, 15, 检测结果分别见表 3 ∼ 6. 本文选用从攻击到正确检测到异常的时间以及误报率作为评价的标准.

从检测时间来看, 在各种攻击情况下检测模型检测到异常的时间随着 *intervals* 的变化而变化. 从检测最坏的结果对应的数据集来看 (对应的数据集为 Dataset4, 此时 $c = 0$, $k = 0.01$), 当 *intervals* 为 5 和 8 时候, 分别在第 436 和第 411 个样本点检测到异常 (在第 20 小时进行注入攻击, 对应的第一个异常样本点为第 401 个). 而当 *intervals* 为 10 和 15 时, 检测模型在第 401 个样本点即检测到异常, 说

表 3　*intervals* = 5 时的检测结果

Table 3　Detection results when *intervals* is equal to 5

| | Dataset1 | Dataset2 | Dataset3 | Dataset4 | Dataset5 | Dataset6 |
|---|---|---|---|---|---|---|
| 正确检测到异常的样本点数 | 409 | 401 | 401 | 436 | 407 | 402 |
| 异常类别 | 异常 2 | 异常 2 | 异常 2 | 异常 2 | 异常 2 | 异常 2 |
| 误报率 (%) | 0.63 | 0.21 | 0.42 | 0.21 | 0.83 | 0.42 |

表 4　*intervals* = 8 时的检测结果

Table 4　Detection results when *intervals* is equal to 8

| | Dataset1 | Dataset2 | Dataset3 | Dataset4 | Dataset5 | Dataset6 |
|---|---|---|---|---|---|---|
| 正确检测到异常的样本点数 | 401 | 401 | 401 | 411 | 401 | 401 |
| 异常类别 | 异常 2 | 异常 2 | 异常 2 | 异常 2 | 异常 2 | 异常 2 |
| 误报率 (%) | 5.62 | 4.38 | 5.21 | 4.17 | 5.83 | 5.42 |

表 5　*intervals* = 10 时的检测结果

Table 5　Detection results when *intervals* is equal to 10

| | Dataset1 | Dataset2 | Dataset3 | Dataset4 | Dataset5 | Dataset6 |
|---|---|---|---|---|---|---|
| 正确检测到异常的样本点数 | 401 | 401 | 401 | 401 | 401 | 401 |
| 异常类别 | 异常 2 | 异常 2 | 异常 2 | 异常 2 | 异常 2 | 异常 2 |
| 误报率 (%) | 7.50 | 7.08 | 8.12 | 8.21 | 6.67 | 7.71 |

表 6 $intervals = 15$ 时的检测结果

Table 6 Detection results when $intervals$ is equal to 15

| | Dataset1 | Dataset2 | Dataset3 | Dataset4 | Dataset5 | Dataset6 |
|---|---|---|---|---|---|---|
| 正确检测到异常的样本点数 | 401 | 401 | 401 | 401 | 401 | 401 |
| 异常类别 | 异常 2 | 异常 1 | 异常 1 | 异常 1 | 异常 1 | 异常 1 |
| 误报率 (%) | 24.37 | 26.25 | 22.92 | 30.21 | 28.96 | 26.04 |

明当 $intervals$ 增大时, 检测模型对异常更加敏感, 检测异常的速度有所提升.

从误报率来看, 当 $intervals$ 逐渐增大时, 相应地误报率也随之增大. 当 $intervals$ 为 5 时误报率均不超过 1 %. 当 $intervals$ 增大到 8 时, 误报率迅速增加到 5 % 左右. 而当 $intervals$ 取 15 时, 误报率超过 20 %, 这时可认为状态迁移图不能正确检测异常.

### 3.5 主元分析法测试

由于工控领域缺乏标准的数据集, 研究人员都用各自的模型和数据进行工控系统异常的研究, 很难复现, 难以对比各种方法的好坏. 为了进一步评估状态迁移图检测的性能, 将其与控制系统常用的异常检测方法 — 主元分析法 (Principal component analysis, PCA) 进行比较. PCA 采用 $T^2$ 统计量和 SPE 统计量作为检测异常的指标, 当样本的 $T^2$ 统计量和 SPE 统计量超过各自的控制限时, 检测到异常. 选定状态迁移图模型检测结果最差的数据集 Dataset4 作为 PCA 方法的测试集, 其测试结果如图 14 和图 15 所示. 从图 14 和图 15 可以看出, PCA 虽然最终能检测到异常, 但是在第 500 个样本点才检测出来. 相比于状态迁移图检测模型, 检测到异常的速度较慢.

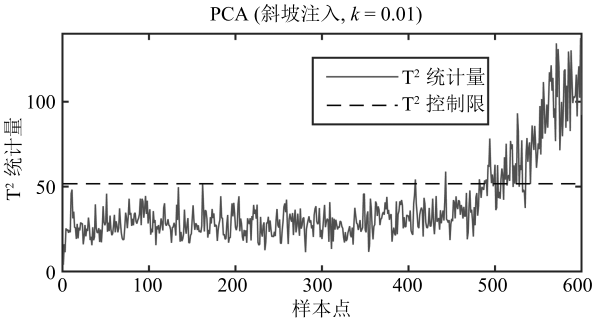



图 14 PCA 方法 $T^2$ 统计量

Fig. 14 $T^2$ statistic of PCA method

### 3.6 讨论

本节从时空资源消耗以及检测性能两方面对状态迁移图检测模型进行讨论, 由于训练过程是通过历史数据离线进行的, 因此时空消耗只考虑检测时的情况.

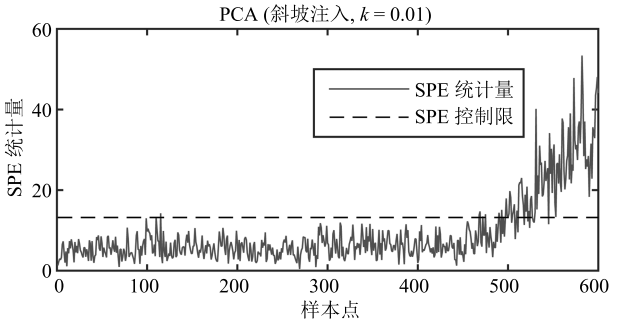



图 15 PCA 方法 SPE 统计量

Fig. 15 SPE statistic of PCA method

1) 从时间消耗上来看, 对新样本点的检测需要判断新样本点是否位于状态迁移图内, 即需要遍历一次状态图, 若状态迁移图含 $n$ 个顶点, 则时间复杂度为 $O(n)$. 实际检测过程中, 顶点数与设置的量化区间数 $intervals$ 有关. 通过第 3.4 节的实验结果不难发现, 当 $intervals$ 取 5 或 8 时已有较好的检测结果, 继续增大 $intervals$ 只会增大资源消耗, 同时增加误报率. 且 $intervals$ 越大, 需要越多的训练样本进行训练. 以 $intervals$ 取 10 为例, 顶点数 $n$ 最多为 100, 而实际会小于 100, 因此这一步骤需要的计算量很小. 第二个步骤判断上一状态到当前状态是否可达, 只需查询邻接矩阵中对应的数值即可, 时间复杂度为 $O(1)$.

2) 从空间消耗上来看, 本文的异常检测方法需要存储量化阈值集合 $\boldsymbol{r}$ 和 $\boldsymbol{d}$, 状态矩阵 $STATE$, 邻接矩阵 $NM$, 其中邻接矩阵 $NM$ 占用的存储空间为 $n$ 的二阶次, $\boldsymbol{r}$ 和 $\boldsymbol{d}$, 状态矩阵 $STATE$ 存储消耗为 $n$ 的一阶次. 但由于 $n$ 取值很小 (不超过 100), 且邻接矩阵存储单位为比特, 当 $intervals$ 取 10 时, 估计整体消耗的存储空间不超过 1 MB.

3) 从检测性能来看, 即便对于微小的攻击 (对应数据集 Dataset1 和 Dataset4), 检测模型也能较快检测出异常, 相比于常规的异常检测算法, 例如第 3.4 节的 PCA 算法, 状态迁移图检测模型能够更好地刻画系统运行的动态变化, 更快地检测出异常. 但是当 $intervals$ 较大 (取 10 以上) 时, 检测误报率较高. 而当 $intervals$ 较小 (例如, 取 5 时) 时, 不

能快速地对微小攻击做出反应. 因此需要谨慎选择 *intervals* 以平衡误报率和检测速度之间的关系.

## 4 总结

本文提出了一种新的基于状态迁移图的异常检测方法, 详细描述了其状态表示、状态图生成和在线检测过程, 建立了工控系统的恶意数据攻击模型, 并基于 TE 仿真模型进行了各项实验, 最后从时空资源消耗和检测性能两方面对方法进行了讨论. 该方法具有较小的时空消耗, 且考虑了系统运行的动态特性, 能够快速检测出异常, 因而适用于资源受限和对实时性要求较高的工业控制系统.

## References


1 Jia Chi-Qian, Feng Dong-Qin. Industrial control system devices security assessment with multi-objective decision. *Acta Automatica Sinica*, 2016, **42**(5): 706−714
(贾驰千, 冯冬芹. 基于多目标决策的工控系统设备安全评估方法研究. 自动化学报, 2016, **42**(5): 706−714)

2 Langner R. Stuxnet: dissecting a cyberwarfare weapon. *IEEE Security and Privacy*, 2011, **9**(3): 49−51

3 Urbina D I, Giraldo J A, Cardenas A A, Tippenhauer N O, Valente J, Faisal M, Ruths J, Candell R, Sandberg H. Limiting the impact of stealthy attacks on industrial control systems. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. Vienna, Austria: ACM, 1994. 1092−1105

4 Rahman M A, Al-Shaer E, Kavasseri R G. A formal model for verifying the impact of stealthy attacks on optimal power flow in power grids. In: Proceedings of the 2014 ACM/IEEE International Conference on Cyber-Physical Systems. Berlin, Germany: IEEE, 2014. 175−186

5 Kiss I, Genge B, Haller P, Sebestyén G. A framework for testing stealthy attacks in energy grids. In: Proceedings of the 2015 IEEE International Conference on Intelligent Computer Communication and Processing. Cluj-Napoca, Romania: IEEE, 2015. 553−560

6 Queiroz C, Mahmood A, Tari Z. A probabilistic model to predict the survivability of SCADA systems. *IEEE Transactions on Industrial Informatics*, 2013, **9**(4): 1975−1985

7 Yoon M K, Ciocarlie G. Communication pattern monitoring: improving the utility of anomaly detection for industrial control systems. In: Proceedings of the 2014 NDSS Workshop on Security of Emerging Networking Technologies. San Diego, California, USA: The Internet Society, 2014.

8 Beaver J M, Borges-Hink R C, Buckner M A. An evaluation of machine learning methods to detect malicious SCADA communications. In: Proceedings of the 12th International Conference on Machine Learning and Applications. Florida, USA: IEEE, 2013. 54−59

9 Maglaras L A, Jiang J M. Intrusion detection in SCADA systems using machine learning techniques. In: Proceedings of the 2014 Science and Information Conference. London, England: IEEE, 2014. 626−631

10 Kroll B, Schaffranek D, Schriegel S, Niggemann O. System modeling based on machine learning for anomaly detection and predictive maintenance in industrial plants. In: Proceedings of the 2014 IEEE Emerging Technology and Factory Automation. Barcelona, Spain: IEEE, 2014. 1−7

11 Stefanidis K, Voyiatzis A G. An HMM-based anomaly detection approach for SCADA systems. *Information Security Theory and Practice*. Cham: Springer-Verlag, 2016. 85−99

12 Shang W L, Zeng P, Wan M, Li L, An P F. Intrusion detection algorithm based on OCSVM in industrial control system. *Security and Communication Networks*, 2016, **9**(10): 1040−1049

13 Khalili A, Sami A. SysDetect: a systematic approach to critical state determination for industrial intrusion detection systems using Apriori algorithm. *Journal of Process Control*, 2015, **32**: 154−160

14 Wang Y, Xu Z Y, Zhang J L, Xu L, Wang H P, Gu G F. SRID: state relation based intrusion detection for false data injection attacks in SCADA. *Computer Security-ESORICS 2014*. Cham: Springer-Verlag, 2014. 401−418

15 Fovino I N, Carcano A, De Lacheze Murel T, Trombetta A, Masera M. Modbus/DNP3 state-based intrusion detection system. In: Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications. Perth, Australia: IEEE, 2010. 729−736

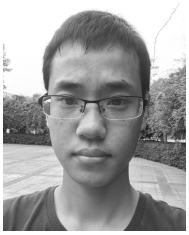
16 Genge B, Siaterlis C, Karopoulos G. Data fusion-base anomay detection in networked critical infrastructures. In: Proceedings of the 43rd Annual IEEE/IFIP Conference on Dependable Systems and Networks Workshop. Budapest, Hungary: IEEE, 2013. 1−8

17 Carcano A, Coletta A, Guglielmi M, Masera M, Fovino I N, Trombetta A. A multidimensional critical state analysis for detecting intrusions in SCADA systems. *IEEE Transactions on Industrial Informatics*, 2011, **7**(2): 179−186

18 Downs J J, Vogel E F. A plant-wide industrial process control problem. *Computers and Chemical Engineering*, 1993, **17**(3): 245−255

19 Ricker N L. Tennessee Eastman challenge archive [Online], available: http://depts.washington.edu/control/LARRY/TE/download.html, May 31, 2017

20 Ricker N L. Decentralized control of the Tennessee Eastman challenge process. *Journal of Process Control*, 1996, **6**(4): 205−221


**吕雪峰** 数学工程与先进计算国家重点实验室硕士研究生. 主要研究方向为工控安全. E-mail: lvxuefeng10@163.com
(**LV Xue-Feng** Master student at the State Key Laboratory of Mathematical Engineering and Advanced Computing. His main research interest is industrial control system security.)



**谢耀滨** 解放军信息工程大学网络空间安全学院讲师. 主要研究方向为工控安全. 本文通信作者.
E-mail: yb_xie@163.com
(**XIE Yao-Bin** Lecturer at the School of Cyber Space Security, PLA Information Engineering University. His main research interest is industrial control system security. Corresponding author of this paper.)