Detecting Semantic Attack in SCADA System: A Behavioral Model Based on Secondary Labeling of States-Duration Evolution Graph

Lijuan Xu, Bailing Wang, Xiaoming Wu, Dawei Zhao, *Member, IEEE*, Lei Zhang, and Zhen Wang, *Senior Member, IEEE*

***Abstract*—By violating semantic constraints that the control process impose, the semantic attack leads the Industry Control Systems (ICS) into an undesirable state or critical state. The spread of semantic attack has caused huge economic losses and casualties to critical infrastructure. Therefore, detecting semantic attack is referred to an urgent and critical task. However, few existing detecting techniques can achieve satisfactory effects in detecting semantic attack of ICS, due to the high requirements of complete critical state-based semantic behavior features description, joint detection on multivariate type state variables, and validity of field states datasets under semantic attacks. In an effort to deal with above challenges, We label device states databases with temporal characteristics and divide impacts on states of field devices under semantic attacks into three categories, including absent in states set, confused sequences, irregular frequency. On this basis, we establish a behavioral model based on secondary labeling of states-duration evolution graph (BMSLS), then implement an adaptive secure state-based semantic attack detection framework furtherly. Compared with the traditional Auto Regression (AR) algorithm, the newer time series correlation graph model, and other five deep learning algorithms, our proposed framework demonstrates the superior effect on the detection of semantic attack.**

***Index Terms*—Critical state, semantic attack, state-based duration evolation graph, supervisory control and data acquisition (SCADA).**

Manuscript received April 8, 2021; revised November 5, 2021; accepted November 22, 2021. Date of publication November 25, 2021; date of current version March 23, 2022. This work was supported in part by the National Key Research and Development Project of China under Grant 2018YFB2004200, in part by the National Natural Science Foundation for Distinguished Young Scholars under Grant 62025602, in part by the National Natural Science Foundation for Distinguished Young Scholars under Grant 62025602, in part by the National Natural Science Foundation of China under Grants 62172244, U1803263, 11931015, and 81961138010, in part by the Frontier Science and Technology Innovation of China under Grant 2016QY05X1002-2, in part by the Shandong Provincial Key Research and Development Program of China under Grant 2019JZZY010132 and in part by the Natural Science Foundation of Shandong Province under Grants ZR2020YQ06 and ZR2021MF132, in part by the Fok Ying-Tong Education Foundation of China under Grant 171105, and in part by the Key Technology Research and Development Program of Science and Technology-Scientific and Technological Innovation Team of Shaanxi Province under Grant 2020TD-013. Recommended for acceptance by Dr Kui Ren. *(Corresponding authors: Bailing Wang; Zhen Wang.)*

Lijuan Xu is with the School of Computer Science and Technology, Harbin Institute of Technology, Weihai 264209, China, and also with the Shandong Provincial Key Laboratory of Computer Networks, Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences), Jinan 250014, China (e-mail: xulj@sdas.org).

Bailing Wang is with the School of Computer Science and Technology, Harbin Institute of Technology, Weihai 264209, China, and also with the Research Institute of Cyberspace Security, Harbin Institute of Technology, Harbin 150001, China (e-mail: wbl@hitwh.edu.cn).

Xiaoming Wu, Dawei Zhao, and Lei Zhang are with the Shandong Provincial Key Laboratory of Computer Networks, Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences), Jinan 250014, China (e-mail: wuxm@sdas.org; zhaodw@sdas.org; zhanglei@sdas.org).

Zhen Wang is with the School of Artificial Intelligence, Optics and Electronics (iOPEN), Northwestern Polytechnical University, Shaanxi 710072, China, also with the School of Cyberspace, Northwestern Polytechnical University, Shaanxi 710072, China, and also with the School of Mechanical Engineering, Northwestern Polytechnical University, Shaanxi 710072, China (e-mail: zhenwang0@gmail.com).

Digital Object Identifier 10.1109/TNSE.2021.3130602

I. Introduction

SEMANTIC attack, also referred to sequence attack, represents a kind of activity that is legal at the protocol level, yet violates semantic constraints that a process imposes, including both semantically incorrect messages (e.g., conflicting commands) and operations that lead the Industry Control System (ICS) into an undesirable state or critical state (e.g., a command to open a pump when it must remain shut) [1]. Traditional attacks are realized by monitoring network and industrial control protocol vulnerability, which is relatively easy to be detected. Semantic attack means that the command conforms to the protocol specification, but violates the production logic process of the industrial control system, which demand sophisticated attackers who have comprehension of industrial control process. Deep packet parsing and statistics of network traffic characteristics could detect traditional attacks with a favorable feasibility. However, in reality, most attack patterns are focused on semantic attack, which are difficult to be detected with traditional detection mechanisms. As a result, the methods to add semantic descriptions into network traffic characteristics, such as parameters relative to control commands or trusted measurement of the sensors, have been proposed to satisfy the new detection requirements. In spite of that, above detection mechanisms are powerless in the situation of compromising field devices directly. On the other hand, attack mechanisms slowly penetrated into ICS are more difficult to detect. Consequently, the semantic attack detection, i.e., recognizing the abnormal valve switching frequency, has become a crucial task for intrusion detection in the SCADA system.

To detect semantic attack effectively, a variety of system critical states-based detection techniques have been proposed in which the concept of critical states and secure states was introduced and defined. And then, a series of techniques, such as developing detection rules [2]–[4], predicting state transition trends [5], [6], and monitoring operation sequences [7], [8], were further exploited. The above research has achieved periodic results, such as constructing description language and secure state-rules, improving attack detection accuracy, and so on. However, there are still some deficiencies to be solved.

- Incomplete critical state-based semantic behavior features description. Industrial State Modeling Language (ISML) and some secure state-rules have been constructed to describe the critical states and secure states of ICS. However, in the real ICS, the definition of critical state-based semantic behavior feature is a lack of comprehension. As a result, without a comprehensive consideration of attack mechanism, this situation would inevitably result in insufficient detection capabilities.
- Joint detection requirements for multivariate type state variables. At present, most state-based detection methods focus on the behavior of separate detection of multidimensional continuous variables or multidimensional discrete variables, without considering both the transitivity of discrete variables and the timing sequence of continuous variables.
- Requirements for validity and credibility of field states datasets under semantic attacks. Public datasets associated with semantic attack, mostly concentrated on network traffic datasets, while temporal state datasets are rare. State-based semantic attacks detection scheme requires a more realistic and effective state dataset.

Motivated by the above, we propose a behavioral model based on secondary labeling of States-duration Evolution Graph (SEG), named BMSLS that label operation mode of field devices in the control system automatically, according to various of device semantic behavior, which are manifested as the duration of the device state and the order relationship between different device states.

Compared to existing secure state-rules construction and model based methods, BMSLS has the following advantages.

- Effective. According by traditional definitions of semantic attack, we divide effects on states of field devices under semantic attacks into three categories, including absent in states set, confused sequences, irregular frequency. Take into account this, we construct SEGs with semantic characteristics to describe the behavior of ICS, implying effective theoretical basis.
- Accurate. Compared with the AR algorithm, time series correlation graph model [9], and other five deep learning algorithm, the proposed algorithm has higher F1 and lower false positive rate.
- Valid. From water distribution system (WDS), whose architecture and performance are in close proximity to the actual water distribution systems, field states datasets reflect the corresponding phenomenon of state changes truthfully. Therefore, the corresponding BMSLS constructing on the basis of datasets is valid.

The remainder of this paper is organized as following. Section II reviews the existing definition of semantic attacks and relative research on intrusion detection method against semantic attacks. Section III explores effects on states of field devices under semantic attacks. Section IV provides our approach in details. Section V describes an experimental analysis using datasets from our developed testbed. Conclusions and suggestions for future work are presented in Section VI.

II. Related Work

A. Semantic Attack

Hadžiosmanović *et al.* [1] considered semantic attack represents the most challenging target for detection. According by the attack pattern, they distinguished three subtypes of process attacks: (i) reconnaissance (e.g., discovering process configuration), (ii) direct control (e.g., modifying process setpoints) and (iii) indirect control (e.g., tampering with exchanged variable values to trigger undesirable reaction). The reconnaissance attack demonstrates that it analyses functionality a Programming Logic Controller (PLC) implements or structure of memory map, for example, probing function code that defines the type of functionality in Modbus, listening for responses and exceptions and probing readable/writable points, and so on. From the above, it can be seen that the reconnaissance attack would not cause the device into an undesirable state that engineers can not sensed. The main purpose of this attack was to assist the attacker to understand the production logic process of the industrial control system. In this paper, we research semantic attacks concentrating on the (ii) and (iii). As early as 1997, an U.S. report on critical infrastructure protection described a sequence attack being the cause of "water hammer effects" [10]. An attacker sent an unusually rapid sequence of legitimate write messages issuing open and close commands to the PLC controlling the major control valves, resulting in a large number of simultaneous main breaks in the pipeline. In addition, Stuxnet [11] has a similar attack principle. Besides, Carcano *et al.* [12] discussed a kind of sequence attack that forced two vales in a specified order, bring the system to a critical state, by sending some legal commands to PLC controlling the valves. Summarize the above events, different from the traditional attack classification, Caselli *et al.* [8] and Yang *et al.* [5] believed that sequence attacks can be divided into order-based and time-based attacks. The former is attack who send messages and commands in an illegal and malicious order. The latter is attack who send messages and commands in an illegal time. After that, Sicard *et al.* [13] proposed four kinds of attack pattern, defined as follows.

- Direct attacks. Quickly lead the system to a critical state in order to damage it.
- Sequential attacks. Degrade the behavior of the ICS and send order/report sequences that gradually lead the system closer to a critical state.

- Temporal attacks. Send orders/reports by violating time constraints (periodicity, timing...).
- Over-soliciting attacks. Target equipment to make it unusable for production by premature wear and tear, causing maintenance or breaking equipment.

Except for that, Zhang *et al.* [6] considered another two types of complex semantic attacks, named branch node and periodic characteristics attack, from the perspective of branch sequence order and sequence execution time interval. In actual fact, branch node and periodic characteristics attack are two special cases of order-based and time-based attacks.

The above classification did not express the actual scenarios perfectly, for instance, "Direct attacks" and "Over-soliciting attacks" referred to the attack effects, and did fail to explain any attack techniques. Meanwhile, "Sequential attacks" and "Temporal attacks" would lead to system to a critical state quickly, such as in the tank system [13], forcing "V3" open and "V5" closed as "Tank T5" is full, "Tank T5" would be overflow. Therefore, the pivotal research point of attack model focused on "attack patterns" or "attack effects" are not sufficient.

After all, as far as we known, there is rarely research of industrial control semantic attack model concentrating on attacking targets, attack patterns, attack effects, and relationship with each other simultaneously. Exploring into the relationship between the above three in depth is conducive to comprehend the semantic features of device states and improving accuracy of industrial Internet anomaly detection based on device states.

B. Detection Methodologies of Sematic Attack

By the technique features, semantic attack detection aggregates the methods into three categories, network traffic-based, Industrial Control Specific Protocol(ICSP)-based, and device state-based. For internal or external traffic characteristics, such as periodicity and autocorrelation, the network traffic-based method discovery the abnormal flow by collaboration of detecting multi-points, without parsing the particular protocol specification. Cheung *et al.* [14] were the first to introduce model-based anomaly detection for SCADA systems, specifically focusing on Modbus protocol traffic. The designed a multi-algorithm intrusion detection appliance for Modbus/Transmission Control Protocol(Modbus/TCP) systems with pattern anomaly recognition capabilities, bayesian analysis of Transmission Control Protocol(TCP) headers and stateful protocol monitoring, complemented with customized Snort rules. In addition, neural network [15], fast fourier change and spectrum analysis [16], Integrated Anomaly Detection [17], and N-gram [18] also implemented ICS traffic detection. ICSP-based method, based on industrial control communication protocol specifications, adopts mature protocol format analysis and state protocol analysis technology to detect changes in the protocol format and protocol state in the message and find abnormal behavior. experts has parsed a variety of commonly used industrial control proprietary protocols, such as Distributed Network Protocol 3 (DNP3) [19], and Modbus [20]. Involved methods include support vector machine (SVM) optimized by particle swarm optimization (PSO) [21], and the rough set theory (RST) combined with SVM [22]. Device state-based method detects anomaly behaviors by defining the normal or abnormal status of the device, judging the trend of state transition, and monitoring the sequence of operations founded on business logic and device operating procedures [23].

The ultimate purpose of a process attack is tantamount to trigger anomalies in the field device's states [1]. Consequently, anomaly detection for device state is an effective strategy of detecting semantic attacks, and is widely used to detect semantic attack in SCADA, which has attracted increasing attention simultaneously. Existing studies of device state-based method focused on four categories, i.e., specification-based, rules-based, sequence-based, and model-based.

1) Specification-Based: To guarantee the normal status of system, each field device would preserve a specification file in the engineering workstation, including definition, authorized Internet Protocol (IP) address, IP ports, and function code of industrial control communication protocol *et al.* With above specification, researchers [24]) allowed to build a model of normal behavior by constructing a corresponding white list to detect intrusion. Obviously, this method could not maintain the efficacy once the specification is comprised.

2) Rules-Based: Early in the study, rules-based method is widely investigated in detecting process attack against ICS. According to developing the rules relative to secure state, researchers considered the state violated rules as critical states to diagnose inconsistencies. Carcano *et al.* [23] first proposed the concept of System Critical States (SCS) and defined a language to describe critical states. Meanwhile, Fovino *et al.* [12] presented a first embryonic Intrusion Detection System (IDS) for SCADA protocol in which the concept of state-based analysis was introduced. Extending the above principles, Carcano *et al.* [2] specifically defined a new formalized language called Industrial State Modeling Language (ISML) and predicted criticality by tracking changes in the distance between the current system state and the critical formulas. However, the architecture, they presented, acquired an exhausted system virtual image, keeps a digital representation of the system of physical state. Besides, SysDetect [3] employed a well-established and iterative data mining algorithm, i.e. Apriori, which can be applied to determine the critical states of industrial processes. However, identifying the critical states at each iteration required expertsòpinions. Mitchell *et al.* [4] considered behavior rules for Wide Area Networks (WANs), Neighborhood Area Networks (NANs) and Home Area Networks (HANs) devices controlling autuators and sensors, and proposed a method to transform behavior rules to a state machine. Almalawi *et al.* [25], [26] presented an intrusion detection approach, which based on a data-driven clustering technique of process parameters to automatically identify the normal and critical states of a given system. With the purpose of advancing the effectiveness of the proposed approach, they extracted proximity-

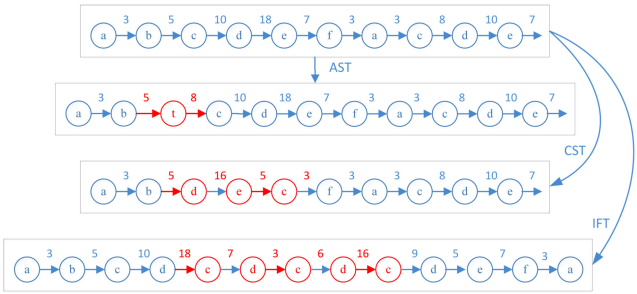


Fig. 1. Effects on states of field devices under semantic attacks.

based detection rules from the identified states. Xu *et al.* [27] used rule mining algorithms and control equipment control logic program protocol algorithm, including Apriori and PrefixSpan, to automatically build the control equipment state rules, which have achieved good effect.

3) Operation Sequence-Based: With the research of deeply progressing, people have discovered that sequence has an extremely essential role in semantic attack. Lin *et al.* [28] combined systematic knowledge of both cyber and physical infrastructure in the power grid to help IDS to estimate execution consequences of control commands. Hadžiosmanović *et al.* [1] developed a semantic and network-based IDS, which extracted process operations from a PLC's raw network packets and constructs a corresponding time series for each process variable to understand its expected activity. Caselli *et al.* [8] proposed a sequence-aware intrusion detection system (S-IDS) reference architecture against semantic attacks, using discrete-time Markov chains(DTMCs), that identifies patterns of ICS network events, extracts their semantic meaning and models known behaviors over time. Khalili *et al.* [7] detected three types of anomalies: anomalous states, anomalous transitions between the normal states, and anomalous time-intervals between the normal transitions. Although they introduced "time-intervals" in state-based intrusion detection, they have no measures facing continuous state data.

4) Model-Based: Auto-association model [29] and hysical process model [30], [31] [32] are typical model-based methods to identify behavior patterns in SCADA systems, which requires a high-detail understanding of physical processes. Markov chains [33], deterministic finite automaton (DFA) for expressing behavior transitions [34] are also utilized for SCADA system and Modbus network. Expect for this, Erez *et al.* [35] developed an automatic classifier to identify three classes of registers: sensor registers, counter registers, and constant registers, and then created a parametritis behavior models for each class. Wang *et al.* [36] proposed a relation-graph-based detection scheme to defeat false data injection attacks on the SCADA system. Despite the fact that low false positive rate is shown in the scheme, it is an unscientific method to judge the relationship between variables, by observing the influence of a change in one variable. Kalech *et al.* [37] proposed cyber-attack detection techniques based on temporal pattern recognition, which based on Hidden Markov Models (HMM) and Artificial Neural Networks (ANN). Ding *et al.* [9] constructed a time series correlation graph model to identify and explain anomalies, using correlation information in multi-dimensional time series data. Above methods rarely considered "time-intervals" characteristic which is a key role in behavior modeling. In order to solve this problem, Yang *et al.* [5] combined the equipment status information to identify the abnormal status of the industrial control equipment in the operation interval to achieve anomaly detection, but they focused on detecting the operation sequence and corresponding state changes without considering the characteristics of state delay. Although Zhang *et al.* [6] measured the state delay characteristics, its method of describing the delay characteristics using the mean value could not express the operating characteristics of the industrial control system thoroughly.

III. The Effect of Semantic Attack on Device States

In order to discover the state-based semantic characteristics of device behavior more comprehensively, we describe the evolution of the device states affected by the attack theoretically.

In most literatures, the field device state refers to the state of the controlled field device read by the sensor, while the output state of the control device(such as PLC, Remote Terminal Unit (RTU)) is regarded as an operation action. However, under certain circumstances, the control behavior is not just affected by the state of the controlled device, but also related to the current behavior of other control devices in a closed control system. Therefore, it is relatively one-sided to simply research the relationship between the control behavior and the controlled device through separating them. We considered system states including not only degrees from sensors connected to controlled devices, but also actions performed by the controlled device(such as valves, pipelines, etc.) after receiving the orders. Extracting semantic characteristics of device behavior through field device state directly, detecting semantic attack further is both reasonable and reliable.

In this paper, according to traditional definitions of semantic attacks, we divide effects on states of field devices under semantic attacks into three categories, including absent in states set, confused sequences, irregular frequency.

- Absent in states set(AST). Order-based attack open/close a valve in an illegal and malicious order, which may trigger system states into an unknown region, never appeared in regular system states.
- Confused sequences(CST). Within a certain period of time, confused sequences arise out of order-based attack changing the switching sequence of two valves or multiple controlled devices in an illegal and malicious order.
- Irregular frequency(IFT). Time-based attacks are sending commands to force field devices to stop the operation currently in progress and proceed to other operations. Although operations of field devices are legal, regular frequency is broken under time-based attacks. The effect of the system state is too short or too long duration time.

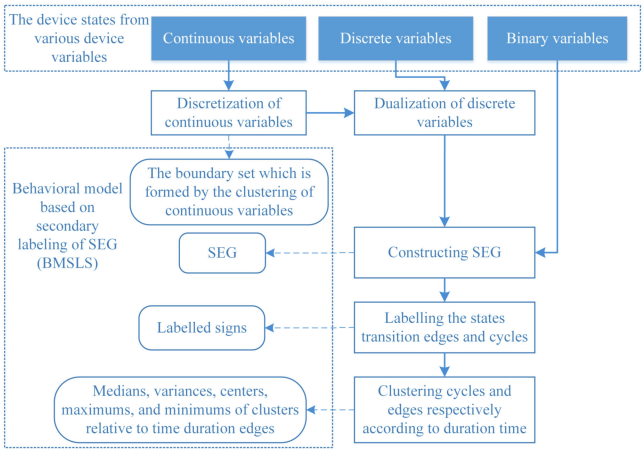


Fig. 2. The detailed construction process of the BMSLS.

Definition 1: System states is $NS = \{(s_i, t_i), i \in [1, N]\}$.

s_i is the state of the control system, while t_i denotes the duration time of s_i.

Fig. 1. is effects on states of field devices under semantic attacks.

IV. Proposed Approach

Fig. 2. is the detailed construction process of the BMSLS.

Firstly, we express all discrete variables and continuous variables as binary type variables respectively to extract device states from multidimensional multivariate type state variables. Secondly, a SEG constructing algorithm is proposed to acquire states transitions and durations. Thirdly, we label cycles and edges without duration characteristics separately in SEG. Lastly, we cluster the labelled cycles and edges according to duration time, and further label cycles and edges with cluster signs. Above all, the BMSLS is comprised of the boundary set which is formed by the clustering of continuous variables, SEG, labelled signs, and medians, variances, centers, maximums, minimums of clusters relative to time duration edges.

A. States-Duration Evolution Graph(SEG)

To reveal effects on states of field devices, we describe semantic characteristics of device variables using the States-duration Evolution Graph. The types of variables that characterize the state of the device are continuous, discrete, and binary. Continuous and discrete variables can also be regarded as binary. Considered that sequentiality is an important feature of device variables, we present semantic characteristics of all binary, discrete variables and a single continuous variable as Definition 2.

Definition 2: States-duration Evolution Graph is: $SEG = (Vs, Ds)$..

$Vs = \{v_0, v_1, \ldots, v_{n_v}\}$ refers to vertex set, representing all device states, and n_v is the number of vertex.

$r_{v_i->v_j}$ denotes a state transition.

$Rs = \{r_{v_i->v_j}, i, j \in [0, n_v]\}$ represents the set of state transitions. v_i and v_j represent vertices in the Vs.

$DT_{v_i->v_j} = \{dt_{v_i->v_{j_1}}, dt_{v_i->v_{j_2}}, \ldots, dt_{v_i->v_{j_{n_t}}}\}$, is the set of durations of a state transition $r_{v_i->v_j}$.

$DTs = \{DT_{v_i->v_j}, i, j \in [0, n_v]\}$ is a set of durations of all state transitions Rs.

$Ds = \{d_1, d_2, \ldots, d_{n_d}\}$ is edges in SEG. In which, n_d is the number of edges. Each edge d_i encompasses a state transition $r_{v_i->v_j}$ and the corresponding set of durations $DT_{v_i->v_j}$.

It is worth mentioning that the edges of the graph not only imply the device state transition relationship, but also preserve the duration of each state transition, which is an essential factor for attack detection.

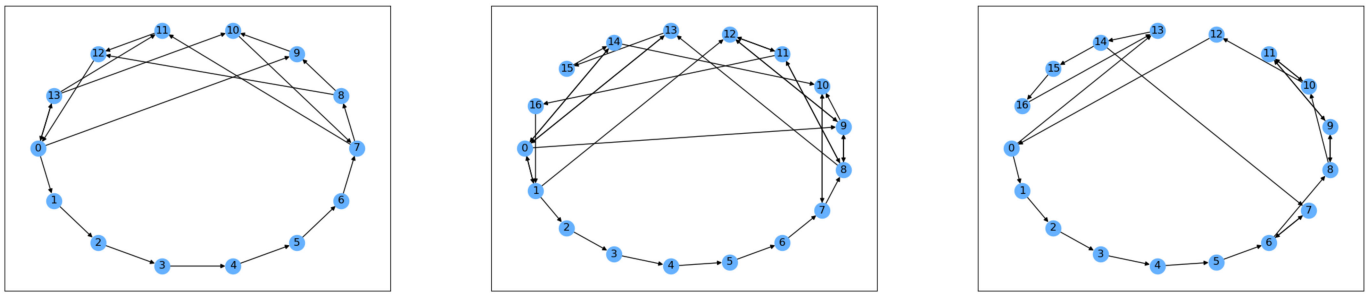
As the intricate relationship between various device variables, combined all types of variables to develop a state transition model is a relatively complicated procedure. Generally, the operating behavior of the execution device is represented by a binary variable. After multiple continuous variables are discredited, the corresponding state space will increase. Therefore, we combine all the binary variables and discrete variables with each continuous variable separately, to build a state-duration evolution graph for detecting semantic attack respectively. In addition, SEG can not expose "order-sequence" of device variables under certain circumstances, such as in the "abcdcdeabdfabc" sequences, "abcd" and "abd" are both correct state transition, "abcd" is at the beginning of sequences, while "abd" is at the middle of sequences. The order of "abcd" and "abd" is indistinguishable in SEG. Therefore, preserving order sequences and time duration sequences are crucial. Furthermore, taking into account the periodicity and continuity of ICS, the order sequences and time duration sequences would be so long that it is sometimes difficult to query. Consequently, labelling sequences of order to time duration characteristics efficiently, and realizing the period of ICS are good solutions.

B. Constructing SEG

The construction of SEG involves the determination of the device states from various device variables, the acquisition of states transitions and durations respectively.

1) Determination of the Device States From Various Device Variables: For extracting device states from multidimensional multivariate type state variables, we express all discrete variables and continuous variables as binary type variables respectively.

- Extraction of device states from discrete variables.
  According to dualization of discrete variables, we extract device states from discrete variables. For instance, a discrete variable D_i with value range $[1, l_d]$, can be expressed as l_d binary variables $\{DB_{i_1}, DB_{i_2}, \ldots, DB_{i_{l_d}}\}$.
  In details, when the value of D_i is $j, j \in [1, l_d]$, the corresponding binary variable DB_{i_j} is 1, and the remaining binary variables $\{DB_{i_1}, DB_{i_2}, \ldots, DB_{i_{j-1}}, DB_{i_{j+1}}, \ldots, DB_{i_{l_d}}\}$ are 0.
- Extraction of device states from continuous variables.
  Extraction of device states from continuous variables consists of two procedures: discretization of continuous



(a) (b) (c)

Fig. 3. Examples of constructed SEGs.

Algorithm 1. Constructing States Evolution Graph

Input: binary dataset DB_b, time dataset DB_t
Output: state set S_s, edges set E_s

$last_l \leftarrow null$ $dura\_step \leftarrow 0$ $beg_i \leftarrow 0$ **for** each line tmp_l in DB_b **do**
  if $last_l == null$ **then**
    $last_l = tmp_l$;
  end
  if $last_l == tmp_l$ **then**
    $dura\_step \leftarrow dura\_step + 1$
  end
  if $last_l \neq tmp_l$ **then**
    $end_i \leftarrow beg_i + dura\_step - 1$;
    $dura\_time = getDuration(beg_i, end_i, DB_t)$;
    if $last_l \subsetneq S_s$ **then**
      Add $last_l$ to S_s;
    end
    $lastl_i = S_s.index(last_l)$;
    if $tmp_l \subsetneq S_s$ **then**
      Add tmp_l to S_s;
    end
    $tmpl_i = S_s.index(tmp_l)$;
    if $lastl_i \rightarrow tmpl_i \notin E_s$ **then**
      $E_s[lastl_i \rightarrow tmpl_i] = [dura\_time]$;
    end
    else
      $E_s[lastl_i \rightarrow tmpl_i].append(dura\_time)$;
    end
    $beg_i = end_i + 1$;
    $dura\_step \leftarrow 0$;
    $last_l = tmp_l$;
  end
end
final;
return;

Algorithm 2. Labelling the states transition edges and cycles

Input: edges set Ds, state transition sequence SS
Output: labelled state transition sequence SS_new, label signs SIG

$CYC \leftarrow null$
$EDG \leftarrow null$
$getCycles(Ds, CYC)$
$sortCYC = sortCycles(CYC)$
$SIG = sortCYC$
for *each item* c_k, c_v in $sortCYC$ **do**
  if $c\_valinSS$ **then**
    SS_new = replaceSign(c_v, c_k, SS);
  end
end
for *each edge* $edge\_v$ in Ds **do**
  if ($edge\_v$ in SS_new) and ($edge\_vnotinDs$) **then**
    $EDG.append(sign\_e, edge\_v)$;
    SS_new = replaceSign($edge\_v, sign\_e, SS\_new$);
  end
end
$SIG.append(EDG)$
final;
return;

variables transforming continuous into discrete variables and dualization of discrete variables, shown as Extraction of device states from discrete variables.

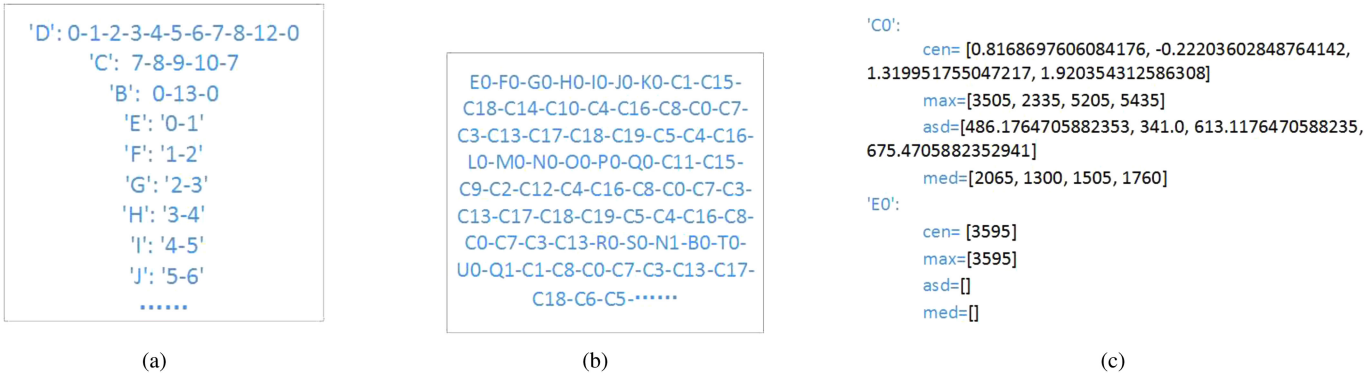
Let $Cs = C_1, C_2, \ldots C_i, \ldots, C_m$ be a set of continuous variables, ranging from l_1 to l_c. In which, C_m is the number of continuous variables. We divide Cs into k clusters $\{CDB_1, CDB_2, \ldots, CDB_k\}$ by clustering algorithm, such as KMeans. The boundary of cluster CDB_1 is $[CD_1, CD_2)$, while the boundary of cluster CDB_2 is $[CD_2, CD_3)$, and so on. The binary variables $\{CB_1, CB_2, \ldots, CB_k\}$ correspond to the clusters $\{CDB_1, CDB_2, \ldots, CDB_k\}$ respectively. When the value of C_i is in range of $[CD_j, CD_{j+1})$, C_i belongs to the cluster CDB_j. In this case, CB_j is 1, and $\{CB_1, CB_2, \ldots, CB_{j-1}, CB_{j+1}, \ldots, CB_k\}$ are 0. According to the analysis above, we conclude that each continuous variable C_i correspond to k binary variables $\{CB_1, CB_2, \ldots, CB_k\}$.

2) SEG Constructing Algorithm: Algorithm 1 describes the particulars of constructing SEG with reprocessed dataset.

Fig. 3 shows examples of constructed SEGs.

C. Behavioral Model Based on Secondary Labeling of SEG

1) Labelling the Sequences of States Transition With Time Duration Characteristics: The duration values $DT_{v_i->v_j}$ corresponding to each state transition $r_{v_i->v_j}$ cause a long states transition sequence for ICS. For instance, in a states transition, we label $r_{v_i->v_j}$ as "a," with duration time 20, $r_{v_j->v_{j+1}}$ as "b," with duration time 10, $r_{v_{j+1}->v_{j+2}}$ as "c," with duration time 5, and so on. In this case, we have a sequence



(a) (b) (c)

Fig. 4. Signs of the states transition edges and cycles with a SEG.

"a20b10c5...". When the next states transition becomes 12,14,21, we get a new sequence "a20b10c5...a12b14c21". As the state transition duration changes, a long sequence of state transitions with duration characteristics is generated during the entire runtime of ICS.

For shortening sequence and further extracting period of SCADA facilitately, two strategies are used to label edges with duration existed in SEG.

Firstly, for all of states transitions Rs, we label cycles and edges without duration characteristics separately in SEG, using graph traversal algorithm. Algorithm 2 describes the particulars of labelling algorithm, in which SS is a temporal state transition sequence composed of states transitions Rs without duration characteristics.

Fig. 4(a) takes an example of labelled the states transition edges and cycles with a SEG shown as Fig. 3(a).

Except for states transition, duration time is another essential element for semantic attack detection. Therefore, we cluster the duration time with cycles and edges signed in the first stage.

Definition 3: Labelled signs: $SN_i, (i \in \{1, 2, \ldots, m\})$, SN_i has $n, (n \geq 1)$ edges. When $n = 1$, SN_i signs a edge. m is the number of labelled signs. A duration time set who exists in a single cycle, corresponding to a labelled sign, is defined as DT_{SN_i}.

$$DT_{SN_i} = \begin{bmatrix} a^i_{1,1} & a^i_{1,2} & \cdots & a^i_{1,p} \\ a^i_{2,1} & a^i_{2,2} & \cdots & a^i_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ a^i_{n,1} & a^i_{n,2} & \cdots & a^i_{n,p} \end{bmatrix} \tag{1}$$

In which, $a^i_{j,k}, (j \in [1, n]), (k \in [1, p])$ is the kth duration time of the jth edge in the labelled sign. p is the number of duration time.

We adopt KMeans++ algorithm to divide duration time set DT_{SN_i} into $q, q \leq p$ clusters.

Formulas (2)–(6) define the particular of the method to normalize DT_{SN_i}, in which, $med^i_j, asd^i_j, max^i_j, min^i_j$ are the median, variance, center, maximum, minimum of the jth edge in SN_i.

$$med^i_j = \begin{cases} (a_{j,p/2} + a_{j,p/2-1})/2, & p > q \ \& \ p\%2 == 0 \\ a_{j,p/2}, & p > q \ \& \ p\%2 == 1 \\ 0, & p = q \end{cases} \tag{2}$$

$$asd^i_j = \begin{cases} 0, & p == q \\ (\sum |a_{j,k} - med_j|)/p, & p > q \end{cases} \tag{3}$$

$$max^i_j = max(a_{j,k}, k \in [1, p]) \tag{4}$$

$$min^i_j = min(a_{j,k}, k \in [1, p]) \tag{5}$$

Formula (6) is the normalization formula, in which, $F(a^i_{j,k})$ is the corresponding normalized value of $a^i_{j,k}$.

$$F(a^i_{j,k}) = \begin{cases} a^i_{j,k}, & p == q \\ (a^i_{j,k} - med^i_j)/asd^i_j, & p > q \end{cases} \tag{6}$$

cen^i_j is center set of the jth edge of q clusters in SN_i, in which, $c^i_{j,r}$ is the center of the jth edge of cluster r in SN_i, and when $p == q$, $c^i_{j,r} = F(a^i_{j,k})$.

$$cen^i_j = \begin{cases} F(a^i_{j,k}), k \in [1, q], & p == q \\ c^i_{j,r}, r \in [1, q], & p > q \end{cases} \tag{7}$$

The median, variance, center, maximum, minimum set of all time duration edges in labelled sign SN_i are denoted as $MED_{SN_i}, ASD_{SN_i}, CEN_{SN_i}, MAX_{SN_i}, MIN_{SN_i}$.

$MED_{SN_i} = \{med^i_1, med^i_2, \ldots, med^i_n\}$,
$ASD_{SN_i} = \{asd^i_1, asd^i_2, \ldots, asd^i_n\}$,
$CEN_{SN_i} = \{cen^i_1, cen^i_2, \ldots, cen^i_n\}$.
$MAX_{SN_i} = \{max^i_1, max^i_2, \ldots, max^i_n\}$,
$MIN_{SN_i} = \{min^i_1, min^i_2, \ldots, min^i_n\}$.

We take the normalized DT_{SN_i} with (6) and the number of clusters p, as input of KMeans++ algorithm. The output of KMeans++ algorithm is cen^i_j. Euclidean distance is used to calculate the distance between vectors in the normalized DT_{SN_i}.

After clustering the cycles and edges, we further label cycles and edges with cluster signs. The combination of signed sequences, label signs, states signs, and cluster

Algorithm 3. Detecting Attacks

Input: $BMSLS$, real-time state data streams tmp_l, continuous variable name $conName$
Output: abnormal dictionary AB_d

$lastStates \leftarrow null$
$duratime \leftarrow null$
$begTime \leftarrow null$
$binVars = getBinaryVars(tmp_l)$
if *JdugeBoundary*($tmp_l[conName], W_s$) == *False* **then**
  Alarm($AB_d[conName]$,curTime,"out of range,"$conName, tmp_l[conName]$);
  continue;
end
tmp_s = discretOfVars($tmp_l[conName], W_s, binVars$);
if $lastStates[conName]$ is *null* **then**
  $lastStates[conName] = tmp_s$;
  $begTime[conName][tmp_s] = curTime$;
  continue;
end
if tmp_s *is not in Vs* **then**
  Alarm($AB_d[conName]$,curTime,"state is unknown,"tmp_s,null);
  continue;
end
if $lastStates[conName] != tmp_s$ **then**
  $duratime[conName][lastStates] = curTime - begTime[conName][lastStates]$;
  expectedInfo = getExpectedInfo($tmp_s, BMSLS$);
  if *JdugeTransition*($lastStates[conName], tmp_s$, $expectedInfo$) == *False* **then**
    Alarm($AB_d[conName]$,curTime,"error transition,"
    $lastStates[conName], tmp_s$);
    continue;
  end
  if *validity*($duratime[conName][lastStates]$, $expectedInfo.cen$) == *False* **then**
    Alarm($AB_d[conName]$,curTime,"lower duration time,"
    $lastStates[conName]$,
    $duratime[conName][lastStates]$);
    continue;
  end
end
final;
return;

information expressing the states transition with duration time characteristics, can represent the normal behavior of the SCADA.

Fig. 3(b) and (c) illustrate the sequence of secondary symbol labeling and the corresponding attributes of each symbol.

Definition 4: Behavioral model based on secondary labeling of SEG (BMSLS) is BMSLS = $SEG, W_s, SIG, MED, ASD, CEN, MAX, MIN$.

In which, SEG is described in **Definition 2**.

W_s is the boundary set formed by the clustering of continuous variables, corresponding to $\{CDB_1, CDB_2, \ldots, CDB_k\}$ in discretization of continuous variables.

SIG is labelled signs, generated by **Algorithm 2**.

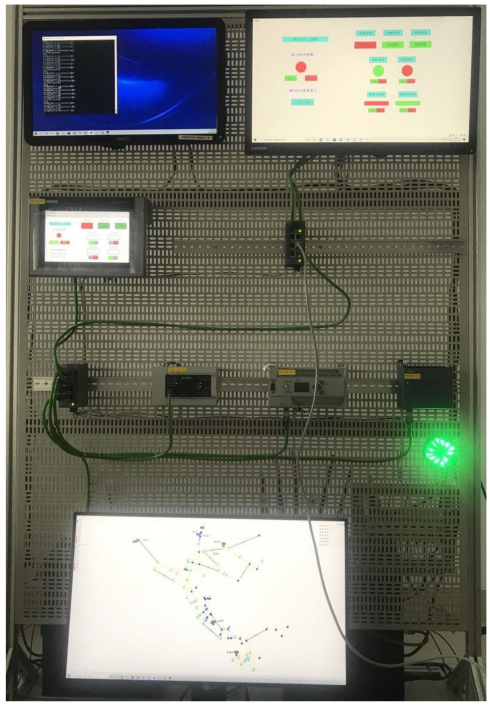


Fig. 5. The actual figure of the implemented combining virtual and real water distribution system.

MED is a set of medians of time duration edges in labelled signs, composed of MED_{SN_i} corresponding to labelled sign SN_i.

ASD is a set of variances of time duration edges in labelled signs, composed of ASD_{SN_i} corresponding to labelled sign SN_i.

CEN is a set of centers of time duration edges in labelled signs, composed of CEN_{SN_i} corresponding to labelled sign SN_i.

MAX is a set of maximums of time duration edges in labelled signs, composed of MAX_{SN_i} corresponding to labelled sign SN_i.

MIN is a set of minimums of time duration edges in labelled signs, composed of MIN_{SN_i} corresponding to labelled sign SN_i.

D. Detection Algorithm

To accurately detect semantic attacks in real time, we input the state data stream generated by the SCADA in real time into established behavioral model based on secondary labeling of SEG. During the detection, we calculate if the duration time is legitimate or not as (8).

$$validity(a^i_{j,k}, c^i_{j,r}) = \begin{cases} |F(a^i_{j,k}) - c^i_{j,r}| < \delta \ and \\ a^i_{j,k} > (1-\epsilon) \cdot min^i_j, & p > q \\ |a^i_{j,k} - c^i_{j,r}| \leq 2 * t_s, & p == q \end{cases} \tag{8}$$

In (8), δ is the variable representing the maximum distance between the center point and the normalized duration time, while ϵ is the coefficient of minimum duration time. t_s is the time interval of receiving state stream.

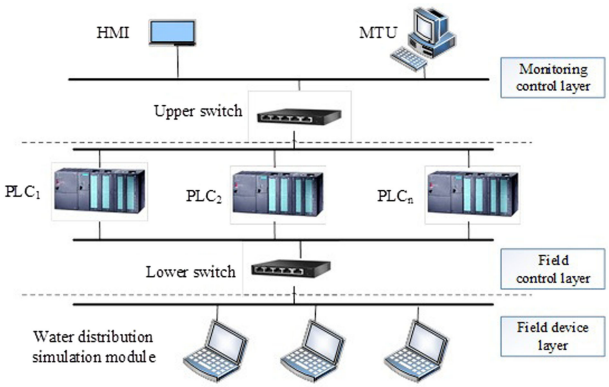


Fig. 6. The schematic figure of the implemented combining virtual and real water distribution system.

Expect for judging the situation of regular state, for instance, the range of a continuous variable, state existence, and state transition legitimacy, detection scheme verifies the correctness of expected state transition time, expected state transition legitimacy as well.

Algorithm 3 describes the particulars of Judging the legitimacy of states and state transitions.

In view of the first attack shortening the expected state transition time of a normal state, if there is no interval between the initial exception duration and the next exception duration, the initial exception delay is pruned directly.

V. Experiments and Analysis

In this section, we present experiments to evaluate the effectiveness of the detection scheme with BMSLS, for detecting semantic attacks on a combining virtual and real water distribution system developed by our team. Figs. 5 and 6. show the actual and schematic figures of the implemented combining virtual and real water distribution system. In the water distribution system, there are three PLCs, who belong to different manufacturers, including Siemens, Mitsubishi, and Rockwell, control pumps, pipelines, and water tankers in the virtual system. Engineer station gathers the levels of water tankers through connected to them, while pumps and pipelines manipulate levels of water tankers to raise or reduce jointly, controlled by the command from PLCs. The system is composed of the target production environment layer, field control layer and field devices layer, including rapid water distribution simulation module, PLC group, Master Terminal Unit (MTU), Human Machine Interface (HMI), and industrial Ethernet communication network and switch connecting each module. The platform can be utilized to industrial control security offensive and defense drills, security product testing, security risk assessment and other related scientific research. By introducing the real-time water delivery and distribution simulation system into the field of industrial control safety testbed, we increase the target scene of the industrial control safety simulation experiment, and solve the difficulties of high cost, time-consuming, and expansion. Substituted virtual PLC, HMI, and MTU, the platform increases the validity and credibility of the source of device states data under semantic attacks, and eliminates the damaged risk of field devices in the attack experiment.

TABLE I
Sample Statistics of the Datasets DB_{nom}, DB_{ai}, and DB_{ac}

Datasets	Samples	Anomalies
DB_{nom}	172801	0
DB_{ai}	20161	2881
DB_{ac}	17281	85

A. Datasets and Attacks on WDS

The dataset DB_{nom}[1] used for constructing BMSLS is gathered from WDS who has run continuously for 10 days in a normal situation, while datasets DB_{ai} and DB_{ac} used for illustrating the analysis effect of detection scheme with BMSLS, is extracted from the same WDS who suffered the order-based attack and time-based attack. The detail introduction of attacks is reproduced below.

- AST & IFT. Continuously sending network packets containing false water level data, to mislead WDS temporarily closing the pipeline because of the water level exceeding the upper limit. PLC can occasionally gather normal water level data, which trigger frequent switching of the pipeline. Finally, above attack consume the service life of the pipeline rapidly, or damage the pipeline directly.
- AST & CST. As soon as MTU was hijacked, a series of sophisticated control commands can be sent to PLC, promoting it to perform the abnormal command, such as shut down the pipeline. Consequently, the stability of the water distribution system disappoints users, owing to the water level does not meet the requirements.

Table I summarises the sample statistics of the datasets DB_{nom}, DB_{ai} and DB_{ac}.

B. Evaluation Measures

We adopt accuracy, precision, false positive rate, false negative rate, and F1 as performance measures. Let $S = \{S_1, S_2, \ldots, S_{n_s}\}$ refer to the set of all states items in the dataset, let $R = \{R_1, R_2, \ldots, R_{n_r}\}$ be the set of the ground-truth attacked states in the states dataset, and let $A = \{A_1, A_2, \ldots, A_{n_a}\}$ be the set of the identified attacked states in the states dataset. we have $n_s > n_r$ and $n_s > n_a$. Let SEG_i refer to the i_th SEGs generated from states dataset. Let TP_i be the identified attacked states correctly according to SEG_i, and the number of SEG is n_e. Let FP_i be the identified attacked states by mistake, let FN_i be the identified normal states by mistake, and let TN_i be the identified normal states correctly. Accuracy, false positive rate, false negative rate, and F1 for all experimental results can be expressed as following.

[1] https://drive.google.com/drive/folders/15WGJObMBu3FULTu7X0fXTkfWMf8cCj13?usp=sharing

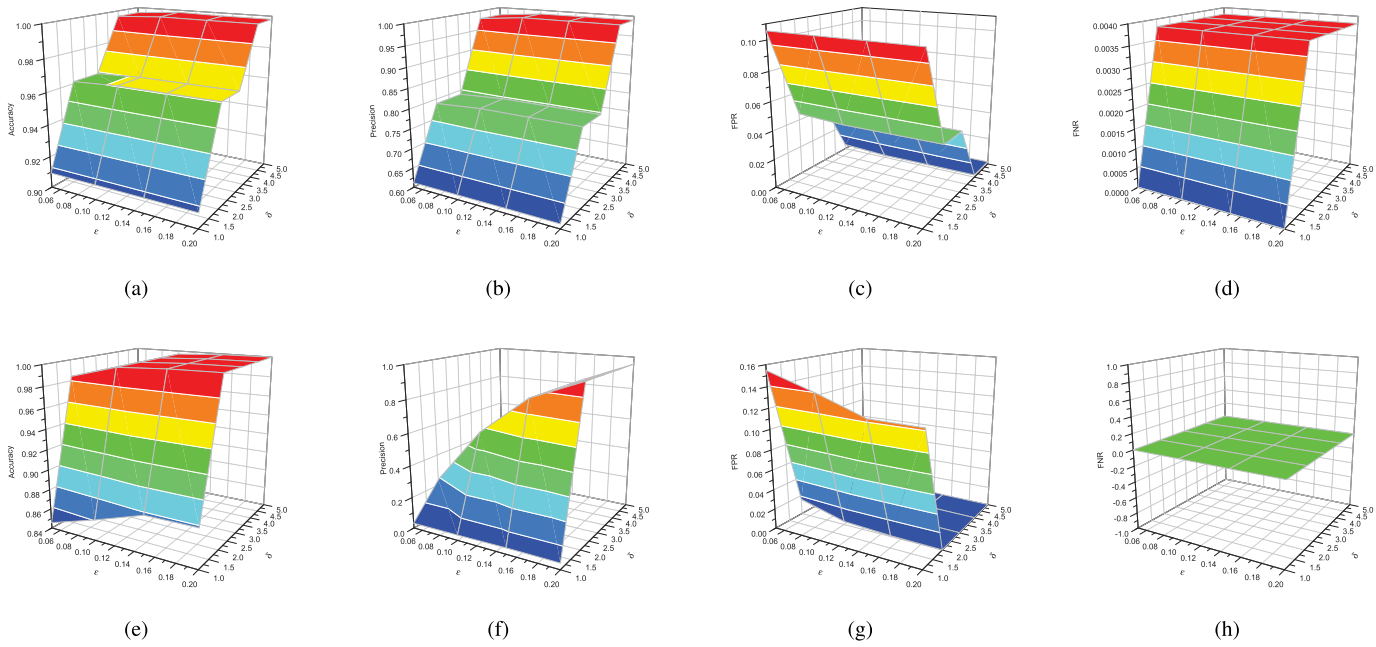


Fig. 7. Joint impact of ϵ and δ on detection performance. Variation in accuracy, false positive rate, and false negative rate for (a)-(d) DB_{ai}, (e)-(h) DB_{ac}.

$$FN = n_r - \bigcup_{n=1}^{n_e} TP_i. \tag{9}$$

$$FP = n_a - \bigcup_{n=1}^{n_e} TP_i. \tag{10}$$

$$TN = n_s - n_r - FP. \tag{11}$$

$$Accuracy = \frac{TP + TN}{n_s} \times 100\%. \tag{12}$$

$$Precision = \frac{TP}{TP + FP} \times 100\%. \tag{13}$$

$$Recall = \frac{TP}{TP + FN} \times 100\%. \tag{14}$$

$$False\ positive\ rate(FPR) = \frac{FP}{FP + TN} \times 100\%. \tag{15}$$

$$False\ negative\ rate(FNR) = \frac{FN}{n_r} \times 100\%. \tag{16}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}. \tag{17}$$

The values of these three measurements vary in the range [0, 1]. A high accuracy value indicates that the detection method can correctly identify most attacks, while a high false negative rate implies that the method identifies fewer attacks than ground-truth. False positive rate measures the overall proportion of attacks identified by mistake. A high F1 value suggests that the method has good balanced precision and recall. Therefore, a high accuracy, a low false negatives, a low false positive, and a high F1 is the objective we expected.

C. Impacts of Detection Parameter δ and τ in (8)

With the discretional parameter $k = 3$ of continuous variables, Fig. 7. shows accuracy, FPR, and FNR, when δ from one to five with a step size of one, and $\epsilon \in [0.05, 0.2]$ with a step size of 0.05in DB_{ai} and DB_{ac}.

TABLE II
NUMBERS OF VERTICES, EDGES, AND CYCLES BELONGING TO DIFFERENT SEGS

Properties	SEGs	K = 2	K = 3	K = 4	K = 5
vertices	SEG_{T1}	12	14	16	21
	SEG_{T2}	12	17	21	25
	SEG_{T3}	12	17	22	26
edges	SEG_{T1}	17	20	21	27
	SEG_{T2}	20	32	38	46
	SEG_{T3}	16	25	33	36
cycles	SEG_{T1}	3	3	3	4
	SEG_{T2}	6	9	13	14
	SEG_{T3}	3	6	8	8

1) In Term of Order-Based Attack, With a Proper ϵ, Increasing the Size of δ Can Improve Detection Performance in Terms of Accuracy: When ϵ is set to a proper value, we find that increasing δ can improve the performance in terms of accuracy. As to $AST\&CST$, the entire accuracy demonstrates a stable appearance. For example, as indicated in Fig. 7(a), the best accuracy is obtained with $\epsilon \geq 0.1, \delta \geq 4$. However, with the expansion of the ϵ and δ range, the FPR has not changed significantly, as shown in Figs. 7(c) and 7(g). The delay of the attacked state brings false positives. For instance, the attacker launches the attack at 20:00, but the state of the field device changes accordingly at 20:00:55. In addition, due to the system state did not immediately return to the original operating track after attack, FNR is generated shown as Figs. 7(d) and 7(h). On the other hand, FNR of dataset DB_{ai} has decreased

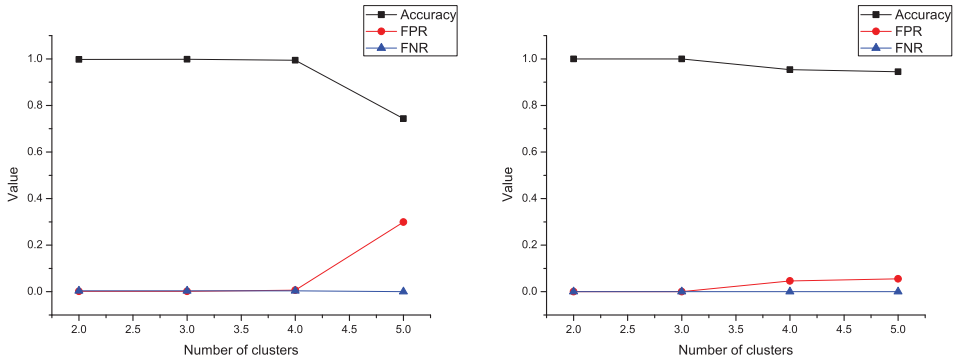


Fig. 8. Joint impact of k on detection performance. Variation in accuracy, false positive rate, and false negative rate for (a) DB_{ai}, (b) DB_{ac}.

TABLE III
A COMPARISON OF THE BENCHMARK ALGORITHMS IN TERMS OF PRECISION, FPR, FNR, AND F1

	DB_{ac}				DB_{ai}			
	Precision	FPR	FNR	F1	Precision	FPR	FNR	F1
AR	5.37	1.64	81.18	0.08	98.99	0.17	0.38	0.99
$TSCGM$	0.00	11.63	100.0	Na	46.37	18.62	3.44	0.63
AE	95.51	0.02	0.0	0.98	100.0	0.0	78.24	0.36
$DAGMM$	100.0	0	54.12	0.63	14.29	100.0	0	0.25
$REBM$	0.58	85.68	0.0	0.01	100.00	0.0	78.24	0.36
$LSTM$	0.78	62.77	0	0.02	14.29	100.0	0.0	0.25
$LSTM-ED$	43.23	0.51	21.18	0.59	100.00	0.0	78.24	0.36
$BMSLS$	95.51	0.02	0.0	0.98	98.99	0.17	0.38	0.99

as the increasing of the size of δ, that led to an increase in accuracy eventually. Therefore, we believe that a proper increase of ϵ and δ would generate an acceptable effect on accuracy, false positives, and false negatives.

D. Impacts of Discretional Parameter k of Continuous Variables

With the proper value of parameters $\epsilon = 0.1$ and $\delta = 4$, we vary the discretional parameter k of continuous variables, which represent the numbers of cluster in the range [2, 5] with a step size of 1. The experimental results of accuracy, FPR, and FNR in DB_{ai} and DB_{ac} are given in Fig. 8.

As the increasing of number of clusters k, the vertices, edges, and cycles of SEGs keep growing, as showed in Table II, which implies that the number of states is increasing with k. Fig. 8(a). shows that the accuracy first increases and then reduce, while FPR demonstrates a decreasing trend. However, with the increasing of k, detection of state transition legitimacy would generate more false negatives. Simultaneously, the expected state transition time is broken after the attack has finished, which will trigger false negatives as well. As a consequence, FNR shows an appearance of first decreasing and then increasing. In conclusion, the performance of detection will not improve with the granularity of clustering on continuous variables.

In Table II, T1, T2, and T3 represent the level of tanker 1, tanker 2, and tanker 3 separately. As to T1, determined by the operating mode of tanker 1, the growth rate in the numbers of vertices and edges is larger than that in the cycles. As growing of k, numbers of vertices and edges of T1, T2, and T3 present an appearance of continued growth. Simultaneously, numbers of cycles are also increasing, but the growth rate is weakening. To further explain, The more meticulous the cluster division is, the less it reflects the operating mode of field devices.

E. Efficiency Analysis and Comparison

In order to objectively verify the performance of the method in this paper, we have used seven time series anomaly detection algorithms as the benchmark algorithms to carry out performance comparison experiments.

The benchmark algorithms are AR algorithm [38], Time series correlation graph model (TSCGM) [9], Autoencoder (AE) [39], Deep Autoencoding Gaussian Mixture Model (DAGMM) [40], Deep Structured Energy Based Models (REBM) [41], Long Short Term Memory Networks(LSTM) [42], and LSTM-based Encoder-Decoder (LSTM-ED) [43].

Table III shows a comparison of the benchmark algorithms in terms of Precision, FPR, FNR, and F1. As shown in Table III, our proposed BMSLS performes better than TSCGM and LSTM in terms of the Precision, FPR, FNR, and F1 on both datasets. The Precision of the DAGMM appears higher than the BMSLS algorithm on dataset DB_{ac}, however, the higher FNR and lower F1 of the DAGMM results in the detection effect of the BMSLS overwhelmed that of the DAGMM. The same phenomenon emerges in the AE, REBM, LSTM-ED, and BMSLS algorithms on dataset DB_{ai}. The AR and BMSLS demonstrate the same Precision, FPR, FNR, and F1 on dataset DB_{ai}, however, the BMSLS exceeds AR on dataset DB_{ac}. Similarly, although it has the same evaluation measures with AE algorithm on dataset DB_{ac}, the BMSLS algorithm exceeds AE on dataset DB_{ai}. We concludes that our proposed BMSLS performs better than other benchmark algorithms.

The appearance of high FNRs on dataset DB_{ac} under AR algorithm is the result of the weak ability of AR to recognize abnormalities on binary variables.

For TSCGM algorithm, no obvious strong correlation between the state variables of training dataset DB_{nom}, which leads to the FNR of dataset DB_{ac} appears 100.00%. Table IV shows the series correlation matrix(SCM) of training dataset DB_{nom}.

T1P, T2P, T3P, P10S, P335S, PIP131S and PIP330S correspond to tankers, pumps, and pipelines in a subsystem of WDS respectively. As showed in Table IV, the maximum series

TABLE IV
SCM OF NORMAL DATASET FROM WDS

	T1P	T2P	T3P	P10S	P335S	PIP131S	PIP330S
T1P	1.000000	0.728782*	0.427570	0.001764**	-0.018301	-0.007916	0.018310
T2P	0.728782*	1.000000	0.372851	0.000034	-0.036149	-0.005557	0.036149
T3P	0.427570	0.372851	1.000000	0.001827	-0.009373	-0.006872	0.009373
P10S	0.001764**	0.000034	0.001827	1.000000	0.00000	0.00000	0.00000
P335S	-0.018301	-0.036149	-0.009373	0.00000	1.000000	0.00000	-0.258258
PIP131S	-0.007916	-0.005557	-0.006872	0.00000	0.00000	1.000000	0.0000
PIP330S	0.018310	0.036149	0.009373	0.00000	-0.258258	0.00000	1.000000

Note: Explanation of correlation; * maximum correlation coefficient; ** minimum correlation coefficient.

correlation coefficient is 0.728782 of Tank1 and Tank2 in DB_{nom}, which reveals not strong correlation between time series of pump 335 and other devices. As a consequence, a higher FNR emerges on the dataset DB_{ac}, created by attacking pump335.

VI. CONCLUSION AND FUTURE WORK

In this paper, we have summarized the semantic attack classification in view of effects of semantic attack on device states, based on this, proposed a detection method, by constructing a behavioral model based on secondary labeling of SEG(BMSLS). To reduce problems caused by joint detection requirements for multivariate type state variables, incomplete definition of semantic characteristics of device relative to secure state, by employing algorithm of establishing states-duration evolution graph and labelling the sequences of states transition with time duration characteristics, we consider combining analysis on the range of continuous variable, state existence, and state transition legitimacy, with on the correctness of expected state transition time, expected state transition legitimacy. Experimental results obtained using a combining virtual and real WDS developed by our team, and two datasets generated from time-based and order-based attacks have demonstrated the effectiveness of BMSLS. It has also been demonstrated that our proposed BMSLS has better performance than other classic algorithms, including AR, TSCGM, AE, DAGMM, REBM, LSTM, and LSTM-ED algorithms.

There are essentially two limitations with this study. First, we did not consider the impact of Kmeans cluster algorithm for discretization of continuous variables on the performance of BMSLS. Second, field devices maybe is still in an abnormal state, although the attack has finished, due to the normal operation trajectories of the field devices have changed. It does not have any ability to distinguish whether the attack has ended yet. In future, as to the first limitation, we will research the influence of different continuous variable discretization segmentation methods on detection efficiency, and explore the optimal segmentation method with deep learning algorithms in deepth. In addition, as to the second limitation, designing more kinds of semantic attack methods and study their impacts on the device state is one of effective solutions. Correlation analysis of the expected impact of the attack and the actual device states transforming trend can achieve accurate detection of abnormal states. There are other aspects we intend to explore in future, including researching probability model of device states changing under semantic attack, and reasoning semantic attack patterns based on above model.

REFERENCES

[1] D. Hadžiosmanović, R. Sommer, E. Zambon, and P. Hartel, "Through the eye of the PLC Semantic security monitoring for industrial processes," in *Proc. 30th Annu. Comput. Secur. Appl. Conf.*, 2014, pp. 126–135.

[2] A. Carcano, A. Coletta, and M. Guglielmi, "A multidimensional critical state analysis for detecting intrusions in SCADA systems," *IEEE Trans. Ind. Informat.*, vol. 7, no. 2, pp. 179–186, May 2011.

[3] A. Khalili and A. Sami, "SysDetect: A systematic approach to critical state determination for industrial intrusion detection systems using apriori algorithm," *J. Process Control*, vol. 32, pp. 154–160, 2015.

[4] R. Mitchell and I. R. Chen, "Behavior-rule based intrusion detection systems for safety critical smart grid applications," *IEEE Trans. Smart Grid*, vol. 4, no. 3, pp. 1254–1263, Sep. 2013.

[5] A. Yang, Y. Hu, and L. Zhou, "An industrial control system anomaly detection algorithm fusion by information flow and state flow," *J. Comput. Res. Develop.*, vol. 055, no. 011, pp. 2532–2542, 2018.

[6] R. Zhang, P. Wu, Y. Lu, and Z. GUO, "Anomaly detection algorithm in ics based on mixed-order markov tree model," *Acta Automatica Sinica*, vol. 46, no. 1, pp. 127–141, 2020.

[7] A. Khalili, A. Sami, A. Khozaei, and S. Pouresmaeeli, "Sids state-based intrusion detection for stage-based cyber physical systems," *Int. J. Crit. Infrastructure Protection*, vol. 22, no. Sep., pp. 113–124, 2018.

[8] M. Caselli, E. Zambon, and F. Kargl, "Sequence-aware intrusion detection in industrial control systems," in *Proc. 1st ACM Workshop Cyber-Phys. Sys. Secur.*, 2021, pp. 13–24.

[9] X. Ding, S. Yu, M. Wang, H. Wang, H. Gao, and D. Yang, "Anomaly detection on industrial time series based on correlation analysis," *J. Softw.*, vol. 31, no. 3, pp. 726–747, 2020.

[10] C. Robert T. Marsh, "C. G. P. O. Staff, Critical foundations: Protecting America's infrastructures, 1998.

[11] C. E. Fallieread N. Murchu LO, "W32.stuxnet dossier," *Symantec Secur. Response*, Version 1.4, Feb. 2011.

[12] I. N. Fovino, A. Carcano, T. D. L. Murel, A. Trombetta, and M. Masera, "Modbus/DNP3 state-based intrusion detection system," in *Proc. 24th IEEE Int. Conf. Adv. Inf. Netw. Appl.*, 2010, pp. 20–13.

[13] Z. E. SicardF. and F. J., "An approach based on behavioral models and critical states distance notion for improving cybersecurity of industrial control systems," *Rel. Eng. Syst. Saf.*, vol. 188, pp. 584–603, 2019.

[14] S. Cheung, B. Dutertre, M. Fong, U. Lindqvist, K. Skinner, and A. Valdes, "Using model-based intrusion detection for SCADA networks," *Proc. SCADA Secur. Sci. Symp.*, 2007, pp. 127–134.

[15] W. Gao, T. Morris, B. Reaves, and D. Richey, "On SCADA control system command and response injection and intrusion detection," *ECrime Researchers Summit*, Oct. 2010, pp. 1–9.

[16] R. R. R. Barbosa, R. Sadre, and A. Pras, "Towards periodicity based anomaly detection in SCADA networks," in *Proc. IEEE 17th Int. Conf. Emerg. Technol. Factory Autom.*, 2012, pp. 1–4.

[17] C. W. Ten, J. Hong, and C. C. Liu, "Anomaly Detection for Cyber security of the Substations," *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 865-873, 2011.

[18] C. Zhou *et al.*, "Design and analysis of multimodel-based anomaly intrusion detection systems in industrial process automation," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 45, no. 10, pp. 1345–1360, Oct. 2017.

[19] T. Mander, F. Nabhani, L. Wang, and R. Cheung, "Data object based security for DNP3 over TCP/IP for increased utility commercial aspects security," in *Proc. IEEE Power Eng. Soc. General Meeting*, 2007, pp. 1–8.

[20] M. Wan, W. Shang, P. Zeng, and J. Zhao, "MODBUS/TCP communication control method based on deep function code inspection," *Inf. Control*, vol. 45, no. 02, pp. 124–132, 2016.

[21] W. Shang, S. Zhang, and M. Wan, "MODBUS/TCP communication anomaly detection based on PSO-SVM," *Acta Electronica Sinica*, vol. 490–491, pp. 1745–1753, 2014.

[22] J. Zhang, P. An, and M. Wan, "Intrusion detection method of RST-SVMfor abnormal behavior in industrial control network," *J. Electron. Meas. Instrum.*, vol. 32, no. 07, pp. 8–14, 2018.

[23] A. Carcano and Fovino, *State-Based Network Intrusion Detection Systems for SCADA Protocols: A Proof of Concept*. Berlin, Germany: Springer, 2010.

[24] R. Sekar, A. K. Gupta, J. Frullo, T. Shanbhag, and S. Zhou, "Specification-based anomaly detection: A new approach for detecting network intrusions," in *Proc. 9th ACM Conf. Comput. Commun. Secur.*, 2002, pp. 265–274.

[25] A. Almalawi, X. Yu, Z. Tari, A. Fahad, and I. Khalil, "An unsupervised anomaly-based detection approach for integrity attacks on SCADA systems," *Comput. Secur.*, vol. 46, pp. 94–110, 2014.

[26] A. Almalawi, A. Fahad, Z. Tari, A. Alamri, R. AlGhamdi, and A. Y. Zomaya, "An efficient data-driven clustering technique to detect attacks in SCADA systems," *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 5, pp. 893–906, May. 2016.

[27] L. Xu, B. Wang, L. Wang, D. Zhao, X. Han, and S. Yang, "PLC-SEIFF: A programmable logic controller security incident forensics framework based on automatic construction of security constraints," *Comput. Secur.*, vol. 92, 2020, Art. no. 101749. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167404820300328

[28] H. Lin, A. Slagell, P. Sauer, and R. Iyer, "Semantic security analysis of SCADA networks to detect malicious control commands in power grids (poster)," in *Proc 7th Int. Conf. Secur. Inf. Netw.*, 2013, pp. 29–34.

[29] D. Yang, A. Usynin, and J. W. Hines, "Anomaly-based intrusion detection for SCADA systems," in *Proc. 5th Int. Topical Meeting Nucl. Plant Instrum. Controls, Human Mach. Interface Technol., Albuquerque*, 2006, pp. 797–802.

[30] N. Svendsen and S. Wolthusen, "Using physical models for anomaly detection in control systems," *IFIP Advances Inf. Commun. Technol.*, Hanover, NH, USA: Springer, vol. 311, 2009, pp. 139–149.

[31] N. Svendsen and S. Wolthusen, "Modeling and detecting anomalies in SCADA systems," in *Proc. Int. Conf. Crit. Infrastructure Protection*, 2008, pp. 101–113.

[32] L. Z. S. Alvaro A. Cárdenas, and S. Amin, "Attacks against process control systems Risk assessment, detection, and response," in *Proc. 6th ACM Symp. Inf. Comput. Commun. Secur.*, 2011, pp. 355–366.

[33] T. Krueger, H. Gascon, N. Kramer, and K. Rieck, "Learning stateful models for network honeypots," in *Proc. 5th ACM Workshop Secur. Artif. Intell.*, 2012, pp. 37–48.

[34] N. Goldenberg and A. Wool, "Accurate modeling of modbus/TCP for intrusion detection in SCADA systems," *Int. J. Crit. Infrastructure Protection*, vol. 6, no. 2, pp. 63–75, 2013.

[35] N. Erez and A. Wool, "Control variable classification, modeling and anomaly detection in modbus/TCP SCADA systems," *Int. J. Crit. Infrastructure Protection*, vol. 102015, Art. no. 1874548215000396.

[36] Y. Wang, Z. Xu, J. Zhang, L. Xu, H. Wang, and G. Gu, "SRID: State relation based intrusion detection for false data injection attacks in SCADA," in *Proc. Eur. Symp. Res. Comput. Secur.*, 2014, pp. 401–418.

[37] M. Kalech, "Cyber-attack detection in SCADA systems using temporal pattern recognition techniques," *Comput. Secur.*, vol. 84, pp. 225–238, 2019.

[38] M. Gupta, G. Jing, and C. Aggarwal, "Outlier Detection for Temporal Data," San Rafael, CA, USA: Morgan & Claypool, 2014.

[39] S. Hawkins, H. He, G. Williams, and R. Baxter, *Outlier Detection Using Replicator Neural Networks*. Berlin, Heidelberg: Springer, 2002, pp. 170–180.

[40] B. Zong *et al.*, "Deep autoencoding gaussian mixture model for unsupervised anomaly detection," in *Proc. 6th Int. Conf. Learn. Representations*, 2018, pp. 1–19.

[41] S. Zhai, Y. Cheng, W. Lu, and Z. Zhang, "Deep structured energy based models for anomaly detection," in *Proc. 33rd Int. Conf. Mach. Learn.*, 2016, pp. 1100–1109.

[42] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, "Long short term memory networks for anomaly detection in time series," in *Proc. 23rd Eur. Symp. Artif. Neural Netw. Comput. Intell. Mach. Learn.*, 2015, pp. 89–94.

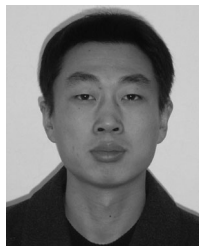
[43] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, and G. Shroff, "LSTM-based encoder-decoder for multi-sensor anomaly detection," 2016, *arXiv:1607.00148*.

Lijuan Xu received the master's degree in computer science and technology from Shandong University, Jinan, China, in 2007. She is currently an Associate Professor with Shandong Computer Science Center (National Supercomputer Center in Jinan), China. Her main research interests include network security, industrial internet security, and computer forensics.

Bailing Wang received his Ph.D. degree from the School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China, in 2006. He is currently a Professor with the School of Computer Science and Technology, Harbin Institute of Technology. His main research interests include financial security, information security, and cyber security.

Xiaoming Wu received the Ph.D. degree in software engineering from the Shandong University of Science and Technology, Qingdao, China, in 2017. He is currently a Professor with Shandong Computer Science Center (National Supercomputer Center in Jinan), China. His main research interests include network security, industrial internet security, and wireless sensor network.

Dawei Zhao (Member, IEEE) received the Ph.D. degree in cryptology from the Beijing University of Posts and Telecommunications, Beijing, China, in 2014. He is currently a Professor with Shandong Computer Science Center (National Supercomputer Center in Jinan), China. His main research interests include network security, complex network, and epidemic spreading dynamics.



Lei Zhang received the master's degree in control theory and control engineering from Shandong University, Jinan, China, in 2005. He is currently an Assistant Professor with Shandong Computer Science Center (National Supercomputer Center in Jinan), China. His main research interests include systems security, industrial control systems (ICS) security, vulnerability mining, and analysis.

Zhen Wang (Senior Member, IEEE) is a Full Professor with Northwestern Polytechnical University, Xi'an, China. He is a Member of Academia Europaea and the European Academy of Sciences and Arts. His current research interests include artificial intelligence, network science, data mining and evolutionary game theory, which have attract over 19000 citations for his works.