JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 33, 1069-1083 (2017)
DOI: 10.6688/JISE.2017.33.4.14

Short Paper

Bayesian Networks for Intrusion Dependency Analysis in Water Controlling Systems

FEIFEI SUN[1], CHAO WU[2] AND DONG SHENG[3]
[1]*Department of Systems Design Engineering*
University of Waterloo
Waterloo, ON N2L 3G1, Canada
E-mail: f556sun@gmail.com
[2]*Science and Technology on Integrated Logistics Support Laboratory*
and College of Mechatronics and Automation
National University of Defense Technology
Changsha, 410073 P.R. China
E-mail: wuchaocnqq@hotmail.com
[3]*Hunan Water Resources and Hydropower Research Institute*
Changsha, 410007 P.R. China
E-mail: 740330071@qq.com

Water controlling systems are important components of smart cities. As a system of Internet of Things, water controlling systems increasingly rely on many heterogeneous perception sensors, transportation tools, and application platforms for providing throughout services. In such environments, dynamic intrusion identification is of crucial to meet security goals and take measures on important nodes considering resources for quality assurance limited by time and by cost. In this paper, we introduce a Bayesian network model for intrusion identification in IoT and propose an importance index to identify important nodes for further security management when intrusion occurs. An experiment is carried out to illustrate how this model works. After formalizing a Bayesian network on a water controlling system, Bayesian inference can be performed based on conditional probability tables of nodes with parents and prior probabilities without parents, which can be acquired statistically based on historical data. This model has good potential applications on Internet of things due to its great capability of coping with thousands of sensors, tools, and platforms with Bayesian inference, and ability of dynamically identifying important nodes to improve the efficiency of security management.

Keywords: Bayesian network, Internet of Things, intrusion detection, importance index, water controlling system

1. INTRODUCTION

Water controlling (WC) plays an increasingly important role in a smart city due to quick urbanization, population growth, and climate change. As a system of Internet of Things (IoT), a WC system, is made up of water supply, precipitation drainage, flood control, wastewater treatment. A WC system inside a smart city is a network with sensor

Received July 15, 2016; revised August 20, 2016; accepted October 11, 2016.
Communicated by Zhe Liu.

nodes that update the quantity and quality of water flow and with arcs that represent the transport channels for water. Inside a flood WC system, many important items are monitored in thousands of monitoring stations: (1) hydrometric includes precipitation, water level, and flow or discharge monitored in many river intersections; (2) mechanical items are related to the structures of river, and reservoirs, such as the structural condition of individual sanitary sewer pipes.

A flood WC system is one of the most important sub-systems to smart cities. So, take a flood WC system as an example to show how a WC system works (Fig. 1). A flood WC system has four elements, including watershed, water dam, river, and city. Ideally, main part of precipitation or melted snow accumulates inside a watershed and flows into a reservoir. A water dam regulates the reservoir. Then water released by the dam goes to a city through rivers.

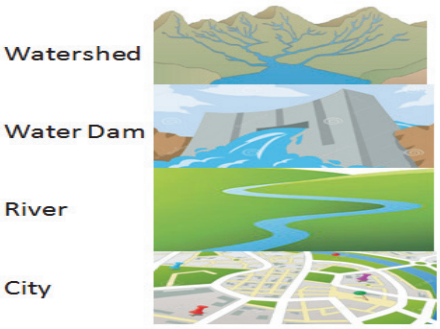

Fig. 1. A flood WC system.

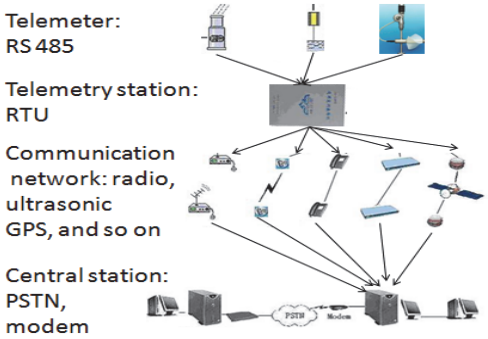

Fig. 2. System architecture for flood control.

The flood WC system architecture is displayed in Fig. 2. It includes telemeters, telemetry station, communication network, and central station. There are different telemeters for hydrometric conditions of watershed and mechanical conditions of reservoirs, rivers, and channels. Listed in the Fig. 2 are rain gauge, laser level meter, and flow meter. Then the generated data is transferred to a telemetry station or a remote terminal unit (RTU) through a RS 485 link. Afterwards the RTU sends the signals to a central station through different communication networks, such as cable network, radio, ultrasonic, GPRS, CDMA, satellite, and so on. For those telemetry stations with ultrasonic communication, it is optional to use relay stations. After receiving the signals, a central station can exchange data with other central stations under Public Switched Telephone Network (PSTN) through a modem.

According to the research on the IoT [1], security architecture of a WC system is summarized (Fig. 3). There are three layers inside, including application, transportation, and perception. Inside the perception layer, there are perception networks made up of perception nodes. As a system of the Internet of Things (IoT), WC systems increasingly rely on thousands of heterogeneous perception sensors, transportation tools, and application platforms for providing throughout services. The more computers are involved, the more malicious usage, attacks, and sabotage are [2]. Further connection with networks such as the Internet and public telephone systems leads to amplified exposure to a variety of attack channels.

Although there are good cryptography methods and cryptographic implementation on sensors [3, 4], cloud search [5], protocols [6], construction methods [7], and authentication [8], information safety in a WC system (IoT) is always a big concern due to these reasons. First, there are many different sensors with different data formats. Second, the on-site sensors are usually too simple to have complicated data security functions. Third, these sensors are commonly located in the outdoors, mountain regions, which let sensors open to attacks. Fourth, there are no special transportation standards for different sensors and so it is impossible to provide unified security protection. Fifth, the transportation layer is based on Internet and current communication networks, whose security threats can affect the security of a WC system.

Application Layer	IoT Application	Smart water controlling: flood control, water supply, precipitation drainage, wastewater treatment.
	Application Support Layer	Middleware technology, service support platform, cloud computing platform, information development platform.
Transportation Layer	Core Network	Internet, and Local area network.
	Access Network	Cable, GPRS, radio, ultrasonic, CDMA, satellite, 3G, PSTN, and so on.
Perception Layer	Perception Network	RFID: protocol; WSN: routing protocol, cryptographic algorithms, key management, and node trust management; RSN: fusion, RFID +WSN;
	Perception Node	Sensor, Base station, Central station, GPRS sim card, RFID Reader, Tag, RTU.

Fig. 3. Security architecture of a water controlling system.

Although wrong or misleading data by deliberate hacks can lead to lots of casualties and property loss by flood, there are only limited resources for assuring security of WC systems by time and cost. First, when a flood is occurring, time is always a limited resource for flood controlling. Second, because there are an extremely large number of facilities involved in a WC system, it is infeasible to provide sufficient cost to ensure the safety of every single facility. So, a smart strategy should be taken to improve the efficiency of maintaining the security of WC systems.

Despite limited time and cost, it is indispensable to take two measures to protect WC systems: intrusion detection and important-node identification. While intrusion detection can help assess the quality of data monitored, important-node identification can help improve cost-effectiveness.

In this paper, we apply Bayesian networks to detect intrusion and identify important nodes in IoT, or WC systems. This paper is organized as follows. Section 2 discusses related work on IoT, dependency analysis, and intrusion detection. Section 3 formalizes the Bayesian model while Section 4 applies the formalized model to a numerical example and discusses the model results. Section 5 makes conclusion on the model and suggest future work.

2. RELATED WORK

Intrusion is any set of actions attempt to compromise the integrity, confidentiality or availability of a data or file. Further, intrusion is categorized into two classes, anomaly intrusions and misuse intrusions [2]. So far, most intrusion detection researches are mainly carried out for network spaces [9-12]. There are three major categories of intrusion detection methodologies: Signature-based Detection, Anomaly-based Detection, and Stateful Protocol Analysis. Essentially, network intrusion detection is to isolate intrusive states from normal states with the help of many methods like neural networks and support vector machines [2], Bayesian networks [13, 14]. Naïve Bayes was applied to detect network intrusion in 2007 and it performed better than a back propagation neural network based approach [13]. Then in 2012 a Hidden Naïve Bayes multiclass classifier was applied for network intrusion detection and the method performed better than other leading state-of-the art models, such as SVM [13, 14].

In comparison to network intrusion detection, Intrusion of the IoT has its own features and challenges [1]. Frist, in perception layer of IoT, the devices have these constraints, including the data rate, small packet (affecting security protocols to achieve the required additional information transmission), limited capacity (8-bit or 16-bit processor architecture, 8 MHz clock frequency), limited energy resources (battery) [1]. Thus, a cost effective intrusion detection method is still an open problem [1]. Second, although the transportation layer of IoT is accessible to attacks, such as Trojan horses, viruses, only necessary and timely intrusion detection mechanisms need to be used [15] due to cost issue. Currently, there are no intrusion detection systems that meet time and cost constraints of IoT because available approaches are either customized for wireless sensor networks or for the conventional internet [16].

Now let's look at dependency analysis. Dependency analysis has been applied a lot in software engineering [17-20], mechanical system [21], and management [22, 23]. A practice-driven systematic review on dependency analysis in software engineering was carried out in 2011 [17]. As summarized in [17], many former researchers in the literature of computer science use the definition of dependency by Stevens et al [24]:

A dependency is the degree to which each component relies on each node of the other components in the software system. The fewer and simpler the connections between components, *the easier it is to understand each component without reference to other components*.

And a good supplement is another definition by Vieira and Richardson that dependencies reflect the potential for one component to affect or be affected by the element of the system. There are three types of dependencies: structural dependencies, behavioral dependencies, and traceability dependencies. These three dependencies can be applied to several areas: 1) application level analysis and management; 2) architecture description and analysis; 3) change impact analysis; 4) system understanding; 5) quality assurances, testing and debugging; 6) refactoring and modularization, 7) traceability and feature analysis.

Many methods have been applied for dependency analysis [22, 25]. Early in 2001, conceptual graphs were applied for dependency analysis [25]. This dependency analysis has two benefits. One is to determine the extent of and impact of a breach in computer systems security or a malfunction in a component while the other is that it is beneficial in

both the requirements and maintenance phases of software engineering. In 2009, fault trees and Bayesian networks were applied for enterprise architecture dependency analysis [22]. This method can be used to evaluate different scenarios. Bayesian networks have been successfully applied into dependency analysis on failure in access control models [26], enterprise architecture [22], cyber security analysis [27].

Considering the constraints of perception devices in IoT, we first introduce Bayesian networks (so far, the best intrusion detection method on internet [22]) to detection intrusion in IoT because this method can only consider two states of a device in IoT, either normal or intrusive, which can meet the resource constraints of IoT in some sense. Second, we also imitate the idea of two papers to identify important nodes to improve cost-effectiveness. One is of using dependency analysis to derive information from service dependencies to measure the relative importance of the service in a service-oriented system [28]. The other is to identify influential factors of business process performance [23]. Differently, we apply Bayesian networks for intrusion detection and use the Bayesian inference results to define importance index of each node.

3. MODELING FORMALISM

3.1 Bayesian Network Formalization

Ahead of all, it is necessary to conceptualize a WC system into a Bayesian network (Fig. 4). As discussed earlier (see Fig. 3), there are three layers in a WC system: perception, transportation, and application. Perception nodes are labeled with three digital numbers while transportation nodes are labeled with two digital numbers. Application nodes are labeled with one digital number. This labeling is open to change when the quantity of nodes involved increases.

Now we describe the theory of Bayesian network for security analysis of a WC system. First, A Bayesian network is a directed acyclic graph where nodes (denoting random variables) are connected by arcs representing probabilistic dependencies [29]. It is worth noting that the networks and nodes inside WC systems are all abstracted as nodes of a Bayesian network $\boldsymbol{B}=(\boldsymbol{N}, \boldsymbol{E})$, where $\boldsymbol{B}$ is a directed acyclic graph. Notes $\boldsymbol{N}$ includes three kinds of nodes: perception node $\boldsymbol{P}$, transportation node $\boldsymbol{T}$, and application node $\boldsymbol{A}$. Edges $\boldsymbol{E}$ represent the conditional probability relationships among nodes. Second, we assume that the perception nodes have no parental nodes and are independent of each other (see an example in Fig. 4). We also need to acquire the prior distributions of each perception node $\boldsymbol{Prob}(\boldsymbol{P})$, and the conditional probability tables (CPT) for each transportation node $\boldsymbol{Prob}(\boldsymbol{T}|\boldsymbol{P})$ and each application node $\boldsymbol{Prob}(\boldsymbol{A}|\boldsymbol{P}, \boldsymbol{T})$. Third, it is assumed that there are only two states for each node, normal **1** and intrusive **2**. The prior distribution and CPT can be acquired from statistics of records through posteriori analysis. Forth, with the conditional independence assumptions [30], the Markov blanket works, which means that a node is only related with its parents and its children and its children's parents.

3.2 Bayesian Inference for Intrusion Detection

With these above, we can do Bayesian inference to assess the conditions of each

node. First, it is straightforward to calculate the marginal distribution of application nodes ***Prob*(*N*)**. Generally there are three methods to calculate the posterior distributions $Prob(n)$, $n \in N$: enumeration, variable elimination, and sampling [31]. The following intrusion detection methods assume that when intrusion is occurring, the dependency (including conditional probability tables) among nodes will change.

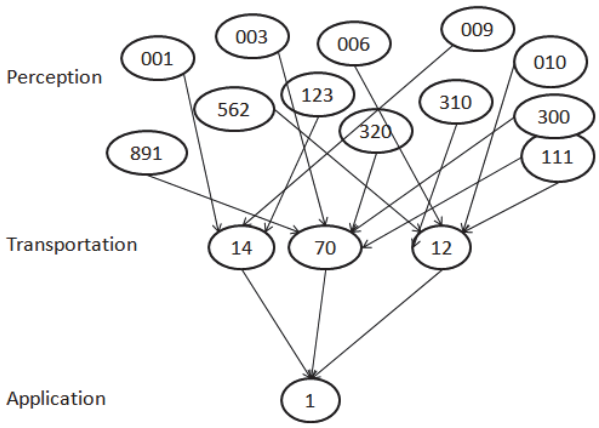

Fig. 4. Bayesian network for water controlling systems. There are three layers in a WC system or IoT: perception, transportation, and application. Perception nodes are labeled with three digital numbers (like 001) while transportation nodes are labeled with two digital numbers (like 70). Application nodes are labeled with one digital number (like 1). This labeling is open to change when the quantity of nodes involved increases.

Here we use the variable elimination for calculation. The algorithm of variable elimination or marginalization eliminates a variable A from a set of Ω of factors, and returns the resulting set of factors. The schematic procedure is as follows:

$\mathcal{F}$=[factors collected relevant to A]
Γ=[the product of all factors in $\mathcal{F}$]
$\mu = \sum_{A} \Gamma$
Return $(\Omega - \Gamma) U[\mu]$

The posterior distribution of each node can be used to detect whether the node was intruded or not. Suppose at time ***t*1**, the posterior distribution of a node is ***Prob*(*t*1)**. Then, later at ***t*2**, after calculation the posterior distribution of the node is ***Prob*(*t*2)**. The comparison between ***Prob*(*t*1)** and ***Prob*(*t*2)** can tell us whether the intrusion on the node occurred or not during the period from ***t*1** to ***t*2**.

Second, with evidence or observation on some nodes ***n*=1 *or* 2**, ***n*∈*N***, we can acquire the conditional probability of any other node conditional on the node or nodes with evidence ***Prob*(*Un*|*Ev*)**, ***Un*∈*N***, or joint probability conditional on evidence ***Prob*(*Mn*|*Ev*)**, ***Mn*∈*N***. ***Un*** refers to a unique unobserved node while ***Mn*** refers to multiple unobserved nodes. ***Ev*** is a set of evidence with a few nodes and their observations, ***Ev*=[*Ev*(1), …, *Ev*(*m*)]**, ***m*∈*N***.

Third, with virtual evidence we can get the probability of other nodes conditional on the virtual evidence ***VEv***. ***VEv*** refers to possible probabilities for each states in which there is no real observations. One is conditional probability of one unique unobserved node on virtual evidence ***Prob*(*Un*|*VEv*)**, ***Un*∈*N***, while the other is conditional probability of multiple unobserved node on virtual evidence ***Prob*(*Mn*|*VEv*), *Mn*∈*N***. ***VEv*** is a set of virtual evidence with a few nodes and their observations, ***VEv*=[*VEv*(1), …, *VEv*(*m*)], *m*∈*N***.

We use junction tree algorithm [32] for calculation of both inference on evidence and virtual evidence. In fact, the junction tree algorithm is a generalization of variable elimination to avoid the query sensitive of variable elimination. The schematic procedure of junction tree algorithm is as follows:

Compile time:

1. Build the junction tree ***T***;
2. Make the density decomposable with respect to ***T***;

Run time:

1. Instantiate evidence in the potentials of the density;
2. Pass messages according to the message passing protocol;
3. Normalize the cluster beliefs/potentials to obtain conditional densities.

For thousands of devices in IoT, it is common to have incomplete security information on some nodes of interested. In this case, first, the conditional probability of a single unobserved node or multiple unobserved nodes on evidence or virtual evidence can provide useful knowledge on their states, either normal or intrusive. Second, the comparison between these conditional probabilities at time ***t*1** and at time ***t*2** can also tell us whether these nodes are attacked or not.

3.2 Importance Index Based on Bayesian Inference

Although intrusive nodes can be detected based on work above, it is still of significance to identify important nodes when intrusion is occurring due to limited cost and time [1]. Since we use Bayesian inference to detection intrusion, the importance index of a node is defined as the difference between the original conditional probability without intrusion and the changed conditional probability with intrusion.

We choose two scenarios to generate the occurrence of an intrusion. One is to represent the condition that there is no intrusion while the other is with an intrusion. The two scenarios can be denoted by two evidences [***Ev*(1), *Ev*(2)**] or two virtual evidence [***VEv*(1), *VEv*(2)**]. Then, with the help of junction tree algorithm we can calculate the probabilities of a single node or nodes. We acquire the difference of probabilities conditional under two scenarios respectively for a single node, ***D*(*n*), *n*∈*N*, *D*(*n*)∈[–1, 1]**. The sign means the direction of how the node is influenced or influencing.

$$D(n) = -|\,Prob(Un \mid Ev(1)) - Prob(Un \mid Ev(2))\,| \tag{1}$$

$$D(n) = -|\,Prob(Un \mid VEv(1)) - Prob(Un \mid VEv(2))\,| \tag{2}$$

We can also calculate the difference of probabilities conditional under two scenari-

os respectively for multiple nodes, $\boldsymbol{D(s)}$, $\boldsymbol{s}=[\boldsymbol{n}, \ldots, \boldsymbol{m}]$, $\boldsymbol{m} \in \boldsymbol{N}$, $\boldsymbol{n} \in \boldsymbol{N}$.

$$D(s) = -|Prob(Mn \mid Ev(1)) - Prob(Mn \mid Ev(2))| \quad (3)$$

$$D(s) = -|Prob(Mn \mid VEv(1)) - Prob(Mn \mid VEv(2))| \quad (4)$$

Then, with these differences, it is easy to arrange them in descending order that means the relative importance of a node or nodes to the security of the whole system under the evidence scenarios, called ***Importance*** index under the evidence change. This index was inspired by these two works [23, 28]. But, we define our own importance index based on Bayesian inference results. The importance sequence has potential application for security resources' allocation to protect important nodes with limiting cost and time.

One way of understanding this ***Importance*** index is like the following. First, consider the evidence change as a kind of intrusion that means the original state of a node or nodes is affected due to intrusion. Second, the ***Importance*** index is one good indicator of expressing how a node or nodes is or are affected by the intrusion.

4. NUMERICAL EXAMPLE

4.1 Problem Description

Now we want to use a numerical experiment to illustrate the application of Bayesian network on intrusion detection and important-node identification for WC systems. With the network (Fig. 5), we have a perception node set ***PN*=[001, 002, 004]**, a transportation node set ***TN*=[03, 05]**, an application node set ***AN*=[6]**. The real problem can be much more complicated than this. It is assumed that the failure of ***PN*** leads to impact the failure of the transportation nodes ***TN***. Further, the failure of ***TN*** can compromise the failure of the application ***AN***. We use these following notations for simplicity: ***PN*=[1, 2, 4]**, ***TN*=[3, 5]**, and ***AN*=[6]**.

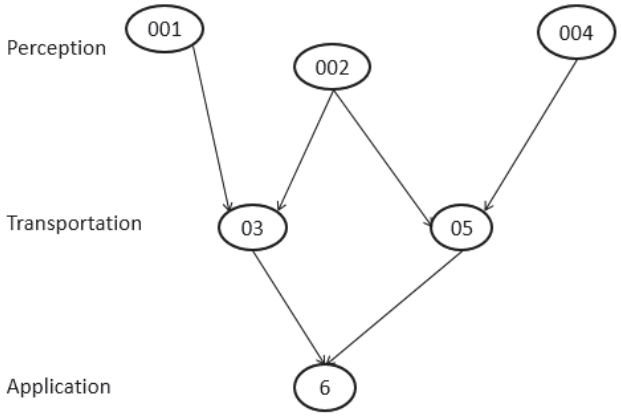

Fig. 5. Bayesian network for the numerical experiment. There are three layers: perception, transportation, and application. Perception layer has nodes 001, 002, and 004. Transportation layer has nodes 03, and 05 while application layer has one node 6.

First, we can have marginal probability tables for each node at time ***t*1.** Then, as time goes to ***t*2,** the prior probability table and conditional probability table can be acquired by doing statistics on historical records between time ***t*1** and time ***t*2** (see Table 1 and Table 2). These data are artificially given, not real observation.

Table 1. The prior probability table for *PN*=[1, 2, 4] at time *t*2.

Node	+ (Normal)	– (Intrusive)
1	0.6	0.4
2	0.3	0.7
4	0.7	0.3

Table 2. The conditional probability table for *TN*=[03, 05] and ***AN*=[06] at time *t*2.**

	Prob(3|1, 2)				Prob(5|2, 4)				Prob(6|3, 5)			
Node	1	2	–3	+3	2	4	–5	+5	3	5	–6	+6
State	–1	–2	0.3	0.7	–2	–4	0.5	0.5	–3	–5	0.3	0.7
	+1	–2	0.1	0.9	+2	–4	0.6	0.4	+3	–5	0.5	0.5
	–1	+2	0.2	0.8	–2	+4	0.3	0.7	–3	+5	0.5	0.5
	+1	+2	0.4	0.6	+2	+4	0.1	0.9	+3	+5	0.4	0.6

4.2 Marginal Distribution

By using the method of variable elimination, we can acquire the marginal distribution for each node at time ***t*2** (in Table 3) based on prior probability (Table 1) and conditional probability (Table 2). It is easy to notice that for perception nodes 1, 2, and 4 their marginal distributions are exactly their prior distributions. Marginal distributions of nodes (3, 5, and 6) give the probability of being normal (not intrusive) or state being + in the following period between time ***t*2** and time ***t*3**. These marginal distributions can be updated dynamically as required.

Table 3. The marginal distribution for each node *Prob*(*n*), *n*∈*N* at time *t*2.

Node		1	2	3	4	5	6
State	–	0.4	0.7	0.222	0.3	0.327	0.4341
	+	0.6	0.3	0.778	0.7	0.673	0.5659

4.3 Conditional Probability on Evidence

Suppose at time ***t*2**, we use two scenarios to artificially generate an intrusion: i) normal and intrusive respectively ***Ev*(1)=[+3, –4]**; and ii) intrusive and intrusive ***Ev*(2)=[–3, –4]**. ***Ev*(1)** is the original states of node 3 and node 4 while ***Ev*(2)** represents the tortured states of node 3 and node 4 after intrusion. Under this condition, it is necessary to locate important nodes to protect considering limited time and resources. So, we make calculation of the conditional probability of on these two evidences at time ***t*2** (Fig. 6). The comparison (in Fig. 6) can tell that the influence led by intrusion or ***Ev*(2)** on different

nodes varies a lot. When ***Ev*(2)** or intrusion happens to nodes 3 and 4, the probability of node 5 being normal (there are two states in each node, either normal or intrusive, defined above) will has the most significant change.

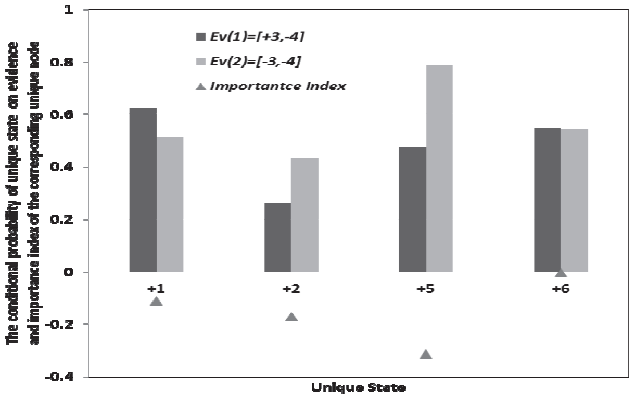

Fig. 6. The conditional probability of unique state onn evidence and importance index of the corresponding unique node.

Further, the importance index can be calculated for each node by Eq. (8). Under the change from original condition ***Ev*(1)** to intrusion condition ***Ev*(2)**, the importance index of node 6 is –0.0047 that means that under this intrusion the node 6 is affected little. Node 5 is mostly influenced with importance index –0.3127. Consequently, node 5 should be protected with priority over other nodes in the following period between time ***t*2** and time ***t*3**, based on the assumption that this intrusion will occur again. These conditional probabilities and important indexes can be updated dynamically as required.

More, similar calculations can be carried out on joint conditional probabilities, such as ***Prob*(+1, –2|*Ev*)**, ***Prob*(+2, +5|*Ev*)**, and ***Prob*(+5, +6|*Ev*)** (in Fig. 7). The comparison under intrusion shows that ***Prob*(+5, +6|*Ev*)** and ***Prob*(+2, –5|*Ev*)** don't change too much, around 0.1. The biggest change occurs to ***Prob*(+1, –2|*Ev*)**, from 0.4859 under ***Ev*(1)** to 0.1892 under ***Ev*(2)**. The seond biggest change happens to ***Prob*(+2+5|*Ev*)**, from 0.1049 under ***Ev*(1)** to 0.3892 under ***Ev*(2)**.

And, importance indexes for multiple joint states can be calculated with Eq. (10). The highest importance index belongs to joint states (+1, –2) while the second highest goes to (+2, +5). Under intrusion on node 3 and 4, (+1, –2) and (+2, +5) are mostly influenced. Consequently, the probability of (+1, –2) and (+2, +5) should be monitored in the following period between time ***t*2** and time ***t*3**, based on the assumption that this intrusion will occur again. These conditional probabilities and important indexes can be updated dynamically as required.

4.4 Probability Conditional on Virtual Evidence

Sometimes the real observations are not available for inference. Then we can use virtual evidence instead to proceed the Bayesian inference. Suppose we have two virtual evidences ***VEv*(1)=[*Prob*(–3)=0.8, *Prob*(+3)=0.2]** and ***VEv*(2)=[*Prob*(–3)=0.2, *Prob*(+3)**

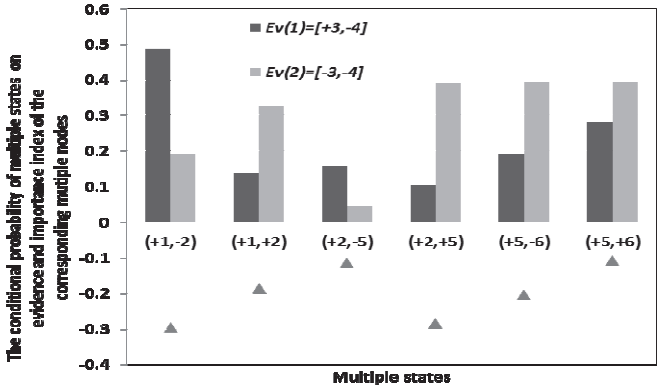

Fig. 7. The conditional probability of multiple states on evidence and importance index of the corresponding multiple nodes.

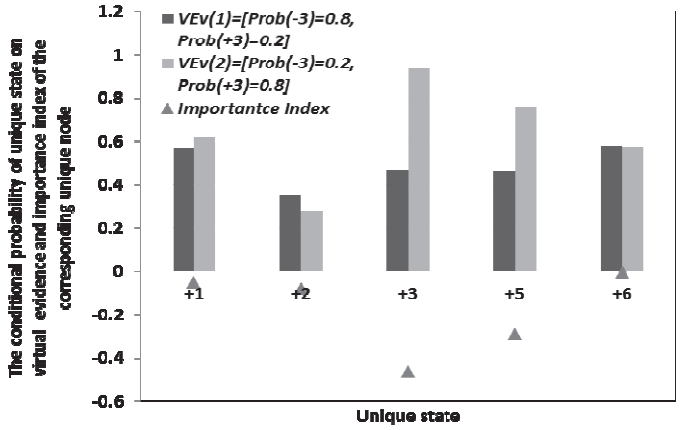

Fig. 8. The joint probability of joint states conditional on evidence.

=0.8]. ***VEv*(1)** is the original states of nodes 3 and 4 while ***VEv*(2)** represents the tortured states of nodes 3 and 4 after intrusion. And we calculate the conditional probability of unique state on these two virtual evidences (in Fig. 8). It is easy to understand that the biggest difference belongs to node 3 since the virtual evidence is for node 3. The second biggest change goes to node 5 of being normal.

The importance indexes for each node show that besides node 3 itself, node 5 is affected the most. When comparing Figs. 8 and 6, it is easy to find similar conditional probability change induced by either evidence or virtual evidence. In other words, evidence and virtual evidence have played similar influences on each node of the Bayesian network.

4.5 Summary

This paper introduces Bayesian network method to a system of IoT, and proposes importance index to differentiate the importance of nodes or multiple joint states. As

discussed [22], Bayesian network methods has these advantages over. First, when the network becomes enormously, Bayesian influence can quickly identify intrusion within short time. Second, Bayesian inference can work with only two states, either normal or intrusive, which is good for coping with constraints of IoT, such as limited capacity, and small packet.

The importance index has many potential applications. One application is that when intrusion occurs, considering limited time and resources it is necessary to choose important nodes or multiple joint states to monitor, not to monitor all nodes of IoT. Because the influence on different nodes by intrusion varies a lot. The monitoring on those nodes affected little by intrusion will play no effect on identifying intrusion and protecting the system of IoT.

5. CONCLUSIONS

As an important of IoT, water controlling systems play a significant role in smart cities. The security of WC systems relies on a large number of perception sensors, transportation tools, and application platforms. It is infeasible to take measurements of all devices to ensure the safety of each component in WC systems due to limited time and resources.

First, we investigate the system and security architectures of water controlling systems. Then, a detailed Bayesian network model is described for the intrusion dependency analysis. Bayesian inference can provide marginal distribution, conditional probability on evidence or virtual evidence. These inferences can be used to determine the extent and impact of an intrusion and to measure the relative importance of the nodes in a water controlling system through an important sequence of the nodes of interest. They are also significantly useful in both the requirements and maintenance phases of security engineering. We demonstrate the application of Bayesian network to intrusion dependency analysis through a numerical example. Our research can promote the security of water controlling systems in smart cities. Future work can focus on how to expand the importance index or put forward other new indexes.

REFERENCES

1. Q. Jing, A. V. Vasilakos, J. Wan, J. Lu, and D. Qiu, "Security of the internet of things: Perspectives and challenges," *Wireless Networks*, Vol. 20, 2014, pp. 2481-2501.
2. S. Mukkamala, G. Janoski, and A. Sung, "Intrusion detection using neural networks and support vector machines," in *Proceedings of International Joint Conference on Neural Networks*, 2002, pp. 1702-1707.
3. Z. Liu, H. Seo, J. Großschädl, and H. Kim, "Efficient implementation of NIST-compliant elliptic curve cryptography for 8-bit AVR-based sensor nodes," *IEEE Transactions on Information Forensics and Security*, Vol. 11, 2016, pp. 1385-1397.
4. Z. Liu, X. Huang, Z. Hu, M. K. Khan, and L. Zhou, "On emerging family of elliptic curves to secure internet of things: ECC comes of age," *IEEE Transactions on Dependable and Secure Computing*, Vol. 14, 2017, pp. 237-248.

5. F. Zhangjie, S. Xingming, L. Qi, Z. Lu, and S. Jiangang, "Achieving efficient cloud search services: multi-keyword ranked search over encrypted cloud data supporting parallel computing," *IEICE Transactions on Communications*, Vol. 98, 2015, pp. 190-200.
6. J. Shen, H.-W. Tan, J. Wang, J.-W. Wang, and S.-Y. Lee, "A novel routing protocol providing good transmission reliability in underwater sensor networks," *Journal of Internet Technology*, Vol. 16, 2015, pp. 171-178.
7. S. Xie and Y. Wang, "Construction of tree network with limited delivery latency in homogeneous wireless sensor networks," *Wireless Personal Communications*, Vol. 78, 2014, pp. 231-246.
8. P. Guo, J. Wang, X. H. Geng, C. S. Kim, and J.-U. Kim, "A variable threshold-value authentication architecture for wireless mesh networks," *Journal of Internet Technology*, Vol. 15, 2014, pp. 929-935.
9. I. Butun, S. D. Morgera, and R. Sankar, "A survey of intrusion detection systems in wireless sensor networks," *IEEE Communications Surveys and Tutorials*, Vol. 16, 2014, pp. 266-282.
10. R. A. Kemmerer and G. Vigna, "Intrusion detection: a brief history and overview," *Computer*, Vol. 35, 2002, pp. 27-30.
11. H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: A comprehensive review," *Journal of Network and Computer Applications*, Vol. 36, 2013, pp. 16-24.
12. C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan, "A survey of intrusion detection techniques in cloud," *Journal of Network and Computer Applications*, Vol. 36, 2013, pp. 42-57.
13. M. Panda and M. R. Patra, "Network intrusion detection using naive bayes," *International Journal of Computer Science and Network Security*, Vol. 7, 2007, pp. 258-263.
14. L. Koc, T. A. Mazzuchi, and S. Sarkani, "A network intrusion detection system based on a Hidden Naïve Bayes multiclass classifier," *Expert Systems with Applications*, Vol. 39, 2012, pp. 13492-13500.
15. L. Zhang and Z. Wang, "Integration of RFID into wireless sensor networks: architectures, opportunities and challenging problems," in *Proceedings of the 5th International Conference on Grid and Cooperative Computing Workshops*, 2006, pp. 463-469.
16. S. Raza, L. Wallgren, and T. Voigt, "SVELTE: Real-time intrusion detection in the Internet of Things," *Ad Hoc Networks*, Vol. 11, 2013, pp. 2661-2674.
17. T. B. C. Arias, P. van der Spek, and P. Avgeriou, "A practice-driven systematic review of dependency analysis solutions," *Empirical Software Engineering*, Vol. 16, 2011, pp. 544-586.
18. T. Armstrong, K. Marriott, P. Schachte, and H. Søndergaard, "Two classes of Boolean functions for dependency analysis," *Science of Computer Programming*, Vol. 31, 1998, pp. 3-45.
19. T. M. Austin and G. S. Sohi, "Dynamic dependency analysis of ordinary programs," in *Proceedings of the 19th Annual International Symposium on Computer Architecture*, 1992, pp. 342-351.
20. C.-Y. Huang and C.-T. Lin, "Software reliability analysis by considering fault dep-

endency and debugging time lag," *IEEE Transactions on Reliability*, Vol. 55, 2006, pp. 436-450.
21. L. Xie, "Pipe segment failure dependency analysis and system failure probability estimation," *International Journal of Pressure Vessels and Piping*, Vol. 75, 1998, pp. 483-488.
22. U. Franke, W. R. Flores, and P. Johnson, "Enterprise architecture dependency analysis using fault trees and bayesian networks," in *Proceedings of the Spring Simulation Multiconference*, 2009, pp. 1-8.
23. B. Wetzstein, P. Leitner, F. Rosenberg, S. Dustdar, and F. Leymann, "Identifying influential factors of business process performance using dependency analysis," *Enterprise Information Systems*, Vol. 5, 2011, pp. 79-98.
24. W. P. Stevens, G. J. Myers, and L. L. Constantine, "Structured design," *IBM Systems Journal*, Vol. 13, 1974, pp. 115-139.
25. L. Cox, H. S. Delugach, and D. Skipper, "Dependency analysis using conceptual graphs," in *Proceedings of the 9th International Conference on Conceptual Structures*, 2001, pp. 117-130.
26. S. S. Alaboodi and G. B. Agnew, "Bayesian networks for modeling failure dependency in access control models," in *World Congress on Internet Security*, 2012, pp. 176-182.
27. P. Xie, J. H. Li, X. Ou, P. Liu, and R. Levy, "Using Bayesian networks for cyber security analysis," in *Proceedings of IEEE/IFIP International Conference on Dependable Systems and Networks*, 2010, pp. 211-220.
28. S. Wang and M. A. Capretz, "Dependency and entropy based impact analysis for service-oriented system evolution," in *Proceedings of IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, 2011, pp. 412-417.
29. E. Gyftodimos and P. A. Flach, "Hierarchical bayesian networks: A probabilistic reasoning model for structured domains," in G. A. Vouros, T. Panayiotopoulos, ed., *Methods and Applications of Artificial Intelligence*, Springer, NY, 2002, pp. 291-300.
30. S. J. Russell, P. Norvig, J. F. Canny, J. M. Malik, and D. D. Edwards, *Artificial Intelligence: A Modern Approach*, Prentice-Hall, NJ, 1995.
31. T. D. Nielsen and F. V. Jensen, *Bayesian Networks and Decision Graphs*, Springer Verlag, NY, 2007.
32. R. G. Cowell, *Probabilistic Networks and Expert Systems: Exact Computational Methods for Bayesian Networks*, Springer Science & Business Media, NY, 2006.

Feifei Sun (孙飞飞) received the B.S and M.S. degrees in Hydrology and Water Resources from Hohai University, China. He is currently pursuing the Ph.D. degree in the Department of Systems Design Engineering, University of Waterloo, Canada. His research interests include optimization under uncertainty, and system reliability.

Chao Wu (吴超) received the B.S. degree from East China University of Science and Technology, Shanghai and M.S. degree from National University of Defense Tech-

nology, China. Both are in Mechanical Engineering. Now he is a Ph.D. candidate in National University of Defense Technology, China. His research interests include system reliability, and fault diagnosis.

Dong Sheng (盛东) received the B.S. and M.S. degrees in Water Supply and Drainage Engineering from Shihezi University, China and received his Ph.D. in Hydrology and Water Resources from Hohai University, China. He is a Senior Engineer in Hunan Water Resources and Hydropower Research Institute, China. His research interests include water resources management, and Hydraulic Safety.