Development of dynamic Bayesian models for web application test management

To cite this article: T V Azarnova et al 2018 J. Phys.: Conf. Ser. 973 012024

View the article online for updates and enhancements.
Development of dynamic Bayesian models for web application test management

T V Azarnova, P V Polukhin, Yu V Bondarenko and I L Kashirina
Department of Applied Mathematics, Informatics and Mechanics, Voronezh State University,
1 Universitetskaya pl., Voronezh, 394006, Russia
E-mail: ivdas92@mail.ru

Abstract. The mathematical apparatus of dynamic Bayesian networks is an effective and technically proven tool that can be used to model complex stochastic dynamic processes. According to the results of the research, mathematical models and methods of dynamic Bayesian networks provide a high coverage of stochastic tasks associated with error testing in multiuser software products operated in a dynamically changing environment. Formalized representation of the discrete test process as a dynamic Bayesian model allows us to organize the logical connection between individual test assets for multiple time slices. This approach gives an opportunity to present testing as a discrete process with set structural components responsible for the generation of test assets. Dynamic Bayesian network-based models allow us to combine in one management area individual units and testing components with different functionalities and a direct influence on each other in the process of comprehensive testing of various groups of computer bugs. The application of the proposed models provides an opportunity to use a consistent approach to formalize test principles and procedures, methods used to treat situational error signs, and methods used to produce analytical conclusions based on test results.

1. Introduction
Modern development of information systems tends to be aimed at integrating technological processes and hardware by means of high-speed communication channels, especially the Internet. This perspective gives an opportunity to solve a wide variety of problems and provides dynamic cluster scalability. Such architecture is built on web applications that simplify structuring information and transmission process organization, and data processing and storage. According to the classical interpretation, web applications are a type of a client-server architecture of information systems with a number of formal characteristics. Cross-platform technologies used for creation and maintenance of web applications provide conditions for their continuous renewal and improvement. Meanwhile specific character of the dynamically changing environment and the tools of application development, which is favorable for development, might give way to stability errors. Testing aimed at early error detection and localization is an inseparable function used at all stages of the architectural design, software implementation, and operation in the context of the information system. The increasing complexity of the internal structure of web applications and the external infrastructure of their operation requires new approaches to testing and modeling of test management processes. Test toolkits based on classical approaches usually require significant resources and are time consuming. At the same time, their capability to detect new errors steadily decreases with time. The fuzzing test method is a flexible, ever-evolving
technology with currently insufficiently developed mechanisms for automated intelligent control. The fuzzing test method is stochastic in nature. To model tools that can be used to manage such processes it is necessary to use models and methods that can reflect the specific character of stochastic processes. Among mathematical modeling methods that can reproduce the specific character of stochastic testing processes by means of the fuzzing method are models based on dynamic Bayesian networks, automata theory, Petri nets, fuzzy and neuron systems, which have a wide range of tools to describe random test variables and parameters and simulation tools to initialize and predict the behavior of the analyzed test process. Different aspects associated with the adaptation of such tools for testing are considered in papers by the following researchers: T.I. Buldakova [1], P.D. Zegzhda [2], I.V. Kotenko [3], A.I. Mischenko [4], P.V. Tumoyan, D.A. Kavchuk [5], etc. For our research, we choose mathematical fuzzing models based on dynamic Bayesian networks due to the high quality of reflection of the test managing tasks in the scope of the formalized possibilities used to describe dynamic processes, to transmit probabilistic information between time slices, and to provide probabilistic inference by dynamic Bayesian networks.

A dynamic Bayesian network is one of the modeling tools used for intellectual data analysis, intended for the deep investigation of complex static and dynamic processes under risk and indeterminateness. In the scientific research, the use of dynamic Bayesian networks to model the process of web application test management will allow us to form a single computational structure that reproduces the functional model of the testing process, describes the logic and probabilistic structure of information and the methodological base of vulnerabilities detection, and implements stochastic intelligent mechanisms of learning, inference, and forecasting in the test procedures. Primary test management findings and results represent results of tasks of filtering, prediction, and smoothing for dynamic Bayesian networks. This paper uses approximate stochastic algorithms and their resource optimization methods to solve the tasks of filtering, prediction, and smoothing.

2. Materials and methods
Testing strategy for specific groups of web application errors is quite specific. It requires a specific setting of the test mechanisms. This article considers the use of dynamic Bayesian networks to test vulnerabilities such as cross-site scripting (XSS) by means of the fuzzing method [9].

The nature of these vulnerabilities is closely related with the client code running in the browser when the web page is loading. In particular, an additional code can be added to the page body without proper authorization and validation. Cross-site scripting covers a wide range of intrusion techniques which can be divided by the code insertion method into the following categories: stored, reflected, and dom [12]. Document object model injection category is associated with the page view in the form of a structured model document and is common for interactive web applications that update data at runtime [13].

The research proposes a conceptual approach to testing that combines methods of random testing and dynamic Bayesian models of testing processes. The random testing by means of the fuzzing method is used to generate samples to be used in: learning structure and probability distributions of individual vertices of a slice, and the formation of evidence for different time slices of the dynamic Bayesian network. Inherently, the testing control process algorithms use a number of formal signs that allow us to determine the structure of such process, as well as to simulate connection handling managerial influence. The following principles can be distinguished: disconnection and feedback. The principle of disconnection is aimed at isolating control signals from external influences, which allows us to block the impact of external factors on the algorithm of the control module. The principle of feedback offers the possibility to correct control action with the help of a correcting signal that signals the lack of effectiveness of a specific functional module that needs adjusting its testing algorithms. The approaches of
self-tuning control models are used to construct test management systems. Such models allow us to get an adaptive and intelligent control system that is able to adjust itself to test a wide range of software errors.

Bayesian models represent directed graphs without cycles, which nodes are modeled as random variables \( X, i = 1,\ldots,n \) with the domains of values \( D_n, i \in D_n \). The arcs leading from the vertex \( X \) to the vertex \( Y \), mean that \( X \in \text{Parents}(Y) \), i.e. \( X \) is the parent node for \( Y \). The vertex \( X_i \) corresponds to the conditional probability distribution \( P(X_i|\text{Parents}(X_i)) \).

The full joint probability distribution for Bayesian networks is as follows \([7, 10]\):

\[
P(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} P(X_i|\text{Parents}(X_i)) \tag{1}\]

Dynamic Bayesian networks represent sequences of Bayesian networks taken in chronological order, connected by logical and probabilistic connections. The vertices of dynamic networks describe the spatial-temporal state of the process under investigation. The set of all vertices of the network \( Z_t = \{X_t, E_t, Y_t\} \) can be distributed into three subsets: \( Y_t \) — hidden variables, \( X_t \) — observation variables, and \( E_t \) — evidence variables.

The mathematical apparatus of static Bayesian networks is aimed at solving the inference problem of the computation of the posterior probabilities \( P(X|E) \) for the set of query variables in the occurrence of some events (evidence) \( e \in E \). The following formula is used to calculate the posterior probability distribution of a Bayesian network \([6]\).

\[
P(X|E) = \alpha \sum_{Z} S_N \omega(X, Z, E), \tag{2}\]

where \( \alpha \) is the normalizing constant.

Interface, brute force, and clustering algorithms are used as basic algorithms to compute \( P(X|E) \). However, in case of the Bayesian networks that model test processes with a high connectivity and a complex structure, these algorithms are inefficient as they are time consuming and their computational cost is quite high. Randomized probabilistic algorithms based on the Monte Carlo method can be used to solve the above-mentioned problem. This family of algorithms is based on the formation and processing of specific samples. The accuracy of the algorithms depends on the method of organization and the dimensions of \( N \) samples, which allows us to find the optimal ratio in terms of adequacy, accuracy, and time. The classical interpretation of the Monte Carlo algorithm is a likelihood weighing sampling algorithm. The main feature of this algorithm is its ability to adjust weight in the process flow of the evidence \( E \) and to reject events inconsistent with the evidence \( e_i \in E_m \). This means that the values of the variables \( E \) are fixed, and the process of formation of the samples is carried out only for the variables \( X \) and \( Z \) (\( S_N = N(X, Z, E) \) — the total number of generated probability samples).

Weights likelihood is computed through recalculation of conditional probabilities for each node of the Bayesian network at the incoming event time \( e_i \) in case of presence of the parent nodes \( \text{Parents}(e_i) \) and can be described by the following expression:

\[
\omega(X, Z, E) = \prod_{i=1}^{m} P(e_i|\text{Parents}(e_i)) \tag{3}\]

The posterior probability distribution in case of all evidences \( E \) is calculated by the formula:

\[
P(X|E) = \alpha \sum_{Z} S_N \omega(X, Z, E) \tag{4}\]

The analysis of the adaptation of the algorithm for testing tasks in the context of dynamic Bayesian networks shows that it is appropriate to combine Monte Carlo and Markov chains,
which makes it possible to tune the dynamical probabilistic inference in temporal models. The mathematical apparatus of Markov chains is used to describe the transitive relations. The transition process is modeled as a first order Markov process \[11\]. The dynamic Bayesian network can be described as a set of Bayesian networks, arranged in chronological order on the time interval \([0,t]\). Let us denote \(X_{t-k,t}, E_{t-k,t}\) is respectively the set of query and evidence variables for time slices during the transition from the time slice \(t-k\) to \(t\). The mathematical model of the dynamic Bayesian network imposes several limitations. The first limitation is associated with conditional probability tables for each of the presented time slices. It means that the table data can be specified once for the 0-th slice, and they do not change when moving from slice to slice. The second limitation is that transitive connections only exist between two neighboring time slices in compliance with the assumption of Markov Rule
\[
P(X_t|X_{0:t-1}) = P(X_t|X_t-1).
\]
The probability distribution \(P(X_{t-1}|X_t)\) describes the transition between the slices and is called a transition model. The state of evidence variable at the time \(t\) is described by the model \(P(E_t|X_{0:t}, E_{0:t-1}) = P(X_t|E_t)\) (it only depends on the current state) \[6\]. The full joint conditional probability distribution for dynamic Bayesian network is as follows:
\[
P(X_n, E_n) = P(X_0) \prod_{i=1}^{n} P(X_{t-1}|X_t)P(E_t|X_t),
\]
where \(P(X_0)\) is the initial probability distribution for zero time.

In case of dynamic Bayesian model for testing of cross-site scripting vulnerabilities, algorithmic components that are used in the error detection process and a set of advanced features to bypass filtering mechanisms and validation parameters can be described as the network nodes. Arcs between vertices in the network determine the probabilistic relationships between these components, both within the slice and the transition between the slices, and determine cause and effect relationships between the algorithms in the context of testing of specific cross-site scripting error groups. Designations for the dynamic Bayesian model for testing cross-site scripting vulnerabilities are given in table 1.

| Table 1. Description of bayesian network nodes |
|----------------------------------------------|
| Node                          | Description                                                      |
| XSS_TYPE                      | Varieties of XSS to embed JavaScript code on a web page.          |
| ENCODER                       | XSS encoding algorithms to overcome software filters.             |
| Evasion                       | Algorithms for XSS obfuscation to overcome software filters.      |
| XSS_PLD                       | Variation of the XSS payload (html tag, event handlers).          |
| KEYLOG                        | Algorithms used to save keyboard pressed keys.                    |
| SPY_EYE                       | Mechanism for obtaining web page screenshots using XSS.           |
| DDOS                          | Mechanism to use web browser as denial of service attack element. |
| PORT SCAN                     | Algorithm for scanning ports on the computer.                     |
| NET_SCAN                      | Scanning the local network, building a network topology.          |
| NAT_PIN                       | Algorithms to bypass NAT traversal rules.                        |
| DR_BY_DW                      | Redirect the user to resources that distribute malware and viruses.|
| BROW_FINPT                    | Overview the components installed in the user’s browser.         |
| AUTHEN,AUTHOR, INTEG,CONF     | Authentication, authorization, integrity, confidentiality mechanisms. |
Figure 1 shows a fragment of a dynamic Bayesian network (two time slices) of the cross-site scripting vulnerabilities testing process. It is important to note that the nodes in the first time slice do not have external relations, and the distribution of conditional probabilities is only determined by the probability \( P(X_0) \). If there is a conditional relation between the variables of two time slices \( P(X_{t-1}) \) and \( P(X_t) \) for two time moments, such a relationship will be constant, otherwise — temporal. Slices of the dynamic Bayesian network describe the state of the test environment and characterize a certain finished step for testing a particular group of applications. Each step is characterized by the processes of calculating parameters of a dynamic Bayesian network (observation and evidence variables) based on the representative data resulting from testing of each error type. Consideration of the testing process in time is, first of all, due to the accumulation of statistical information for each time slice, which offers the possibility to adapt the network to detect anomalous behavior and perform network configuration to identify a certain group of errors [6]. The formation of software error groups is based on the concept of the homogeneity criterion that defines the degree of membership for a particular error by a set of formal characteristics. This enables to generalize some error detection algorithms and optimize the search procedure for the optimal set of test sequences used within the experiment. Obtaining initial distribution, transition, and sensor model also requires initial testing by means of the black box with the subsequent learning of network parameters.

![Figure 1](image.png)

**Figure 1.** The part of dynamic Bayesian network for testing cross site scripting errors.

A fragment of a dynamic Bayesian network shown in figure 1 consists of two layers (slices) of a Bayesian network for the time instants \( t \) and \( t + 1 \) with defined topological relations between the slices that are modeled by means of conditional probabilities between the nodes of the slices. Naive Bayesian classifier is used as a criterion of learning. The mathematical apparatus for
this classifier is based on the maximum posteriori decision rule (MAP) with the following form:

$$\theta_{MAP}^G = \arg\max_\theta \log P(Y|\theta^G) + \log P(\theta^G),$$  \hspace{1cm} (6)$$

where \(\log P(Y|\theta^G) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} N_{i,j,k} \ln \theta_{i,j,k}^G\) is the likelihood log, \(Y = y_1^T, ..., y_N^T\) is the training sample, \(G\) is the structure of the Bayesian network, and \(\theta^G\) is the set of the network parameters.

The Lagrange function is used as the basis for the solution of conditional optimization problem and is represented by the formula:

$$Lagr(G, \theta^G, D) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{i,j,k} \ln \theta_{i,j,k}^G + \sum_{i=1}^{n} \sum_{j=1}^{q_i} u_{i,j} \sum_{k=1}^{r_i} \theta_{i,j,k}^G$$  \hspace{1cm} (7)$$

where \(u_{i,j}\) are the Lagrange coefficients.

The optimization problem can be interpreted as a problem of convex programming, and the necessary and sufficient condition for the maximization of the likelihood log is equality to zero of partial derivatives of the Lagrange function:

$$\frac{\partial}{\partial \theta_{i,j,k}^G} Lagr(G, \theta^G, D) = \frac{N_{i,j,k}}{\theta_{i,j,k}^G} + u_{i,j} = 0 \Rightarrow \frac{N_{i,j,k}}{\theta_{i,j,k}^G} = -u_{i,j},$$  \hspace{1cm} (8)$$

\forall k \Rightarrow \sum_{k=1}^{r_i} = -u_{i,j} \sum_{k=1}^{r_i} \theta_{i,j,k}^G \Rightarrow N_{i,j} := \sum_{k=1}^{r_i} N_{i,j,k} = -u_{i,j}$$

where \(N_{i,j,k} = \sum_{l=1}^{n} N_{i,j,k,l} = \{l \in \{1..N\}|Parents(X_l) = j\}\)

The parameters of the probabilistic distributions of the Bayesian network are as follows:

$$\frac{N_{i,j,k}}{\theta_{i,j,k}^G} = N_{i,j}, \forall k \Rightarrow \theta_{i,j,k}^G = \frac{N_{i,j,k}}{N_{i,j}}$$  \hspace{1cm} (9)$$

The network training procedure is used to fill up the conditional probabilities tables and to calculate the initial probability distribution \(P(X_0)\), the transition \(P(X_{t+1}|X_t)\), and the sensor \(P(X_t|E_t)\) models.

Adaptation of dynamic Bayesian networks to the testing process allows us to solve a number of tasks aimed at detecting anomalous behavior and predicting occurrence probability of error groups by solving the main tasks of probabilistic inference: filtering \(P(X_t|E_{1:t})\), predictions \(P(X_{t+k}|E_{1:t})\), and smoothing \(P(X_t|E_{1:t})\). To address these tasks the research uses a variation of an algorithm for probabilistic inference based on likelihood weighing, i.e. Rao-Blackwell particle filtering algorithm. This algorithm involves the formation of samples in the process of network deployment with their subsequent likelihood weighting. In classical particle filtering algorithm \(N\) samples are formed from the initial distribution \(P(X_0)\), then samples are updated with the transition model \(P(X_{t-1}|X_t)\) during the transition step from the state \(t-1\) to \(t\). The total number of samples for the state \(X_t\) is determined by the following [14]:

$$N(X_{t+1}|E_{1:t}) = \sum_{X_t} P(X_{t+1}|X_t)N(X_t|E_{1:t}),$$  \hspace{1cm} (10)$$

$$N(X_{t+1}|E_{1:t+1})/N = \alpha W(X_{t+1}|E_{1:t+1}) = P(X_{t+1}|E_{1:t+1}),$$

where \(W(X_{t+1}|E_{1:t+1}) = P(E_{t+1}|X_{t+1})N(X_{t+1}|E_{1:t})\) - total weight of the samples in \(X_{t+1}\).

The structure of the particle filtering algorithm is as follows:
The calculation of the conditional probability distributions $P(X_{0:t}|Y_{1:t}, R_{0:t})$ is performed analytically. Estimation of $P(R_{0:t}|Y_{0:t})$ is based on the following expression:

$$P(R_{0:t}|Y_{0:t}) = \frac{P(Y_{t}|Y_{1:t-1}, R_{0:t})P(R_{t}|R_{t-1})P(R_{0:t-1}|Y_{1:t-1})}{P(Y_{t}|Y_{1:t-1})}$$

(12)

Adaptation of the Rao-Blackwell theorem to the topology of the dynamic Bayesian network is based on the comparison criterion of marks (weights) of $N$ samples generated in the process of basic particle filtration algorithm. The mathematical representation of this comparison is shaped by the following inequality [8]

$$\text{var}_{Q(R_{0:t}|e_{t:t})}(\omega(R_{0:t})) \leq \text{var}_{Q(R_{0:t}|X_{0:t}|E_{t:t})}(\omega(R_{0:t}, X_{0:t}))$$

(13)

3. Results and discussion

The research involves a computational experiment in testing cross-site scripting errors. For the purpose of the experiment, a special software was developed and a test environment was created. The test environment represents a system that consists of the following components: a web server with an installed set of web applications with different criticality levels of software errors; the database server PostgreSQL for testing a stored XSS; and the browser factories: Firefox, Chrome, Internet Explorer, Edge. Structurally, the experiment process can be divided into four main phases. The first phase involves a primary testing of applications by means of random fuzzing (white and black box methods). The next phase uses the results of the previous stage to build the Bayesian network and to implement parameter learning procedures: the initial distribution $P(X_0)$, the transition model $P(X_{t-1}|X_t)$, the sensor model $P(X_t|E_t)$, and the time moments $t - 1$ and $t$. The third phase of the experiment involves probabilistic inference procedure.
aimed at obtaining samples that characterize the most typical test assets that can be used to test new applications and detect anomalous behavior within web applications. The final phase is dedicated to the evaluation of the quality of the experiment and the adequacy of test assets to error detection criteria when testing a number of new applications from the homogeneous group.

The proposed mathematical algorithms and models are used to develop the general model of testing cross-site scripting errors, to optimize Rao-Blackwell particle filtering algorithm by parallelizing individual blocks of the algorithm, aimed at the creation of samples and calculating sufficient statistics for the variables of a dynamic Bayesian network. A generalized block diagram of this algorithm is shown in figure 2.

Evaluation of the effectiveness of the proposed algorithm involves a comparison of the developed algorithm and the classical particle filtering algorithm. The comparison is shown in figure 3. It is based on the analysis of two different algorithms within the same computing environment.

The use of parallel operations together with the Rao-Blackwell theorem allows us to optimize the algorithmic base of probabilistic inference as applicable to Bayesian networks with a complex structure without losing precision and effectiveness. At the same time, the testing process becomes more sustainable and adaptive.

The results of the research, supported by computational experiments allow us to speak about the effectiveness of using in testing procedures testing tools that were created as a result of combining of random testing and the mathematical techniques of dynamic Bayesian networks, which are capable of self-training and predicting abnormal errors. The application of this approach to building complex test systems contributes to the effectiveness of solving test management tasks by formalizing relations between the individual elements of testing and by building a clear approach to the timely correction based on the analytical data.
4. Conclusion
The paper considers issues of modeling testing processes and their management based on dynamic Bayesian models. It describes dynamic Bayesian models for the testing of cross-site scripting vulnerabilities. The mathematical apparatus of Bayesian networks allowed us to perform step-by-step modeling, to implement all testing phases, to build structural links between the individual test blocks, and to optimize the existing approaches used in testing. In order to improve the resource efficiency, it was proposed to adapt the Rao-Blackwell theorem to the particle filtering algorithm and to use the parallelizing of individual blocks of the algorithm responsible for the formation of randomized samples. By solving the problems of filtering, prediction and smoothing, the strategy of adaptive static and dynamic web application analysis was successfully implemented to optimize the detection of software errors in testing web applications on different time slices. Comparative analysis with the classical approaches demonstrates the effectiveness of the proposed algorithm and software. The detailed modeling of test management processes by the fuzzing method allows us: to make timely correction of data without direct adjustment of the functional principles of the individual testing modules and to normalize the process of conversion of test chains in logical inference procedures with simultaneous correction of the individual units responsible for test data generation.

5. References
[1] Buldakova T I and Djahalov A H 2013 J. Science And Innovation B 11 1
[2] Zegzhda P D, Kort S S and Suprun A F 2015 J. Automatic Control and Computer Sciences B 49 820
[3] Kotenko D I, Kotenko I V and Saenko I B 2012 J. SPIIRAS Proceedings B 2 5
[4] Mischenko A I 2013 *J. Systems and Means of Informatics* B 1 27
[5] Tumoyan E P and Kavchuk D A 2013 *J. Information Technologies Security* B 1 25
[6] Azarnova T V, Barkalov S A and Polukhin P V 2017 *J. Of Ural State University. Computer technologies, control and radio electronics* B 2(17) 51
[7] Tulup'ev A I 2009 *Bayesian belief networks: logical-probabilistic inference in acyclic direct graphs* (Saint-Petersburg:Saint-Petersburg State University Press)
[8] Suhov Y and Kelbert M 2005 *Probability and Stochastic by Example. Volume 1: Basic Probability and Statistic* (New York: Cambridge University Press)
[9] Beizer B 1995 *Black-Box Testing* (New York: Wiley Publishing)
[10] Murphy K P 2012 *Machine Learning. A Probabilistic Perspective* (Massachusetts: The MIT Press)
[11] Perl J 2009 *Causality: Models, Reasoning and Inference* (New York: Cambridge University Press)
[12] Zalewsky M 2012 *A guide to securing web Applications* (San Francisco: No Starch Press)
[13] Baloch R 2015 *Ethnical Hacking and Penetration Testing Guide* (Boca Raton: CRC Press)
[14] Russel S and Norvig P 2010 *Artificial Intelligence A Modern Approach* (New Jersey: Prentice Hall)