ЖУРНАЛ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И МАТЕМАТИЧЕСКОЙ ФИЗИКИ, 2021, том 61, № 7, с. 1125–1136

ИНФОРМАТИКА

УДК 519.72

ДИНАМИЧЕСКИЕ БАЙЕСОВСКИЕ СЕТИ КАК ИНСТРУМЕНТ ТЕСТИРОВАНИЯ ВЕБ-ПРИЛОЖЕНИЙ МЕТОДОМ ФАЗЗИНГА

© 2021 г. Т. В. Азарнова[1,*], П. В. Полухин[1]

[1] *394018 Воронеж, Университетская пл., 1, ВГУ, Россия*

**e-mail: ivdas92@mail.ru*

Поступила в редакцию 26.11.2020 г.
Переработанный вариант 26.11.2020 г.
Принята к публикации 11.03.2021 г.

В работе рассмотрены вопросы моделирования процессов тестирования веб-приложений методом фаззинга с помощью динамических байесовских сетей. Сформулированы основные принципы оптимизации структуры анализируемых динамических байесовских сетей и предложены гибридные алгоритмы обучения и вероятностного вывода с использованием квазиньютоновских алгоритмов и элементов теории достаточных статистик. Библ. 12. Фиг. 3. Табл. 3.

Ключевые слова: динамические байесовские сети, марковский процесс, критерий Шварца, вероятностный вывод, многочастичный фильтр, критерий условной независимости, теорема Рао–Блекквелла–Колмогорова, алгоритм Левенберга–Марквардта, метод Бройдена.

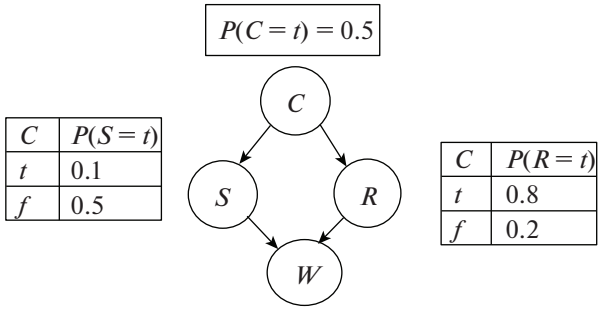
DOI: 10.31857/S004446692107005X

1. ВВЕДЕНИЕ

Предметная область, связанная с разработкой веб-приложений, является достаточно обширной и включает в себя широкий спектр направлений, касающихся проектирования, тестирования и сопровождения веб-приложений. В настоящее время веб-приложения реализуют огромное количество научных, инженерных задач, и успешно используются в процессе организации различных бизнес-процессов. Большинство средств виртуализации, облачных платформ и средств хранения больших данных строятся по архитектуре клиент-сервер и осуществляют механизмы удаленного доступа и управления с помощью сетевых служб и протоколов. Веб-приложения открывают значительные преимущества как для целевых пользователей, так и для создающих их компаний, позволяя систематизировать и самостоятельно производить поддержку и обновление различных функциональных элементов. Несмотря на целый ряд преимуществ использования веб-приложений, они имеют и проблемные зоны. Процесс разработки и функционирования веб-приложений и сервисов сопряжен с возникновением ошибок различного уровня критичности. Многопользовательская эксплуатация веб-приложений может приводить к неконтролируемым потокам входной информации, которые способны вызывать непредсказуемые последствия для устойчивости и безопасности функционирования веб-приложений, в частности, создавать угрозы раскрытия и утечки конфиденциальной информации. Аспекты, связанные с вопросами информационной безопасности, находят широкое отражение в работах российских и зарубежных исследователей, создаются специализированные компании, работающие в области обеспечения безопасности веб-приложений. Особую роль играют открытые проекты, в частности OWASP, специализирующиеся на классификации ошибок, выработке рекомендаций и проектировании механизмов блокирования угроз безопасности веб-приложений. Среди наиболее критичных программных ошибок выделяют: инъекции, межсайтовый скриптинг (XSS) и обход систем аутентификации и управления сессиями. Инъекции классифицируются по типу используемой программной реализации: SQL, инъекции команд и кода [1]. Межсайтовый скриптинг классифицируется по способу внедрения JavaScript кода в веб-страницу – хранимые, отраженные и dom. Механизмы обхода системы аутентификации и управления сессиями реализуют двухэтапные подходы, комбинируя другие методы обнаружения уязвимостей, в особенности, инъекции и межсайтовый скриптинг.

Среди методов обнаружения рассмотренных классов уязвимостей веб-приложений широкое распространение получили методы на основе сканирования, заключающиеся в формировании

специальных тестовых шаблонов. Однако данные методы обладают рядом существенных недостатков, а именно они не позволяют находить аномальные ошибки. Тестирование методом фаззинга было предложено Б. Миллером в качестве альтернативного метода, позволяющего устранить ряд недостатков метода сканирования. Сущность фаззинга [2] заключается в формировании случайного набора тестовых выборок с целью вызова в целевом приложении события сбоя или ошибки и организации мониторинга за реакцией приложения для определения места возникновения анализируемой ошибки. Фаззинг достаточно часто сопоставляют с методом анализа граничных значений, задающим некоторую область допустимых для приложений параметров, и способным отслеживать значения, выходящие за пределы допустимой области. Структурно выделяют несколько групп фаззинга в зависимости от методов тестирования, а также формирования тестовых выборок. С точки зрения формирования выборок, выделяют порождающий и мутационный фаззинг. Основное отличие заключается в том, что мутационный фаззинг осуществляет более осмысленное формирование выборок на базе определенных знаний о структуре или функциях приложения. Среди методов фаззинга выделяют процедуры белого, черного и серого ящиков. Метод белого ящика наиболее применим для обнаружения ошибок, связанных с логикой выполнения программ, обработкой параметров, может использоваться как составной элемент системы анализа покрытия кода тестами. Данный метод широко применяется в процессе решения задач статического анализа кода, используется в системах непрерывной разработки в качестве основного элемента тестирования и поиска программных ошибок. Основной подход, используемый в методе черного ящика, позволяет производить анализ приложения без наличия информации о его внутренних механизмах функционирования. В процессе тестирования методом черного ящика используются случайные генерации наборов входных данных. При этом от тестировщика необходимо лишь иметь общее представление о механизмах функционирования приложения, а также определить набор входных параметров. Существует несколько основных разновидностей тестирования методом черного ящика: эквивалентное разбиение, анализ граничных значений, отладка переходных состояний, функциональные диаграммы, тестирование всех пар значений. Метод серого ящика представляет собой синтез двух описанных выше методов тестирования. Применение данного метода позволяет производить тестирование приложений с частичной информацией о логике обработки параметров, передаваемых между отдельными модулями или подпрограммами. В качестве таких подпрограмм выступают компоненты, предоставляющие механизмы работы с данными, а также реализующие процедуру взаимодействия между процессами и сервисами. С точки зрения моделирования процесса тестирования, тестирование методом серого ящика может быть представлено в виде модели черного ящика с элементами обратной связи. В данном случае обратная связь позволяет формировать управляющие воздействия и своевременно обновлять поток тестовых данных, передаваемый в качестве входных данных рассматриваемой модели, используя ретроспективный анализ. Такой подход позволяет оптимизировать процедуру формирования только тех сценариев, которые позволяют обнаруживать определенные классы ошибок с максимальной вероятностью. Процесс фаззинга серого ящика зачастую носит разделенный во времени характер с поэтапным накоплением статистической информации. Процедуру фаззинга веб-приложений можно представить в виде дискретного или непрерывного стохастического процесса, разделенного на несколько временных срезов. Временные срезы соответствуют периодичности накопления обучающей выборки, используемой для формирования тестовых последовательностей. Применение стохастических процессов для моделирования тестирования методом фаззинга позволяет осуществлять прогнозирование возникновения определенных ошибок при подаче на вход различных типов тестовых данных, а также производить адаптацию тестов для анализа определенного класса программных ошибок за счет применения алгоритмов сглаживания. Эффективным и хорошо апробированным инструментом моделирования стохастических процессов являются динамические байесовские сети (ДБС). Аппарат ДБС включает целый комплекс процедур структуризации, математических методов и алгоритмов, позволяющих моделировать процедуры тестирования методом фаззинга в виде комплексного и адаптивного стохастического процесса. В данной работе рассматриваются различные аспекты применения ДБС для моделирования процессов тестирования веб-приложений методом фаззинга: проектирование ДБС для тестирования определенных уязвимостей, обучения их структуры и параметров, построения процедур вероятностного вывода для достижения целей тестирования. Для реализации процедуры обучения ДБС на нескольких временных срезах используются свидетельства, накопленные до текущего временного среза, формирование транзитивных связей производится на основе проверки гипотез об условной независимости.



C	$P(S=t)$
t	0.1
f	0.5

C	$P(R=t)$
t	0.8
f	0.2

C	R	$P(W=t)$
t	t	0.99
t	f	0.9
f	t	0.9
f	f	0

Фиг. 1. Структура БС с ТУВ (t – истина, f – ложь).

2. ГИБРИДНЫЕ АЛГОРИТМЫ ОБУЧЕНИЯ ДИНАМИЧЕСКИХ БАЙЕСОВСКИХ СЕТЕЙ

Динамические байесовские сети представляют собой разновидность временных моделей и могут быть представлены в виде совокупности статических байесовских сетей (БС), моделирующих исследуемый процесс на определенном временном отрезке. Процедура представления ДБС в виде набора БС является развертыванием сети. БС представляет собой ориентированный ациклический граф с множеством вершин X, представляющих собой дискретные или непрерывные случайные величины, и с множеством дуг G, отражающих отношение родитель–потомок [3]. В качестве множества родительских вершин для множества вершин Y рассматривается множество

$$\text{Parents}(Y) = \{x : \exists y \in Y, \exists(x, y) \in G\}.$$

Множество детей для множества вершин Y определяется в виде:

$$\text{Children}(Y) = \{x : \exists y \in Y, \exists(y, x) \in G\}.$$

В теории статических байесовских сетей принято использовать топологическую нумерацию, при которой родительские вершины получают меньшие номера, чем дети. Существование топологической нумерации для любой байесовской сети доказано. Каждая вершина БС характеризуется таблицей условных вероятностей (ТУВ). Пример простейшей БС приведен на фиг. 1.

При исследовании байесовских сетей важнейшую роль играют гипотезы об условной независимости, факторизации, разделении. Гипотеза об условной независимости представляет собой предположение, что каждый узел y при известных значениях родителей $\text{Parents}(y)$ не зависит от любого множества X, такого, что $x \notin Y$ и $X \subsetneq Y$. Формальное представление гипотезы об условной независимости для байесовской сети имеет следующий вид:

$$P(x_i, y|\text{Parents}(x_i)) = P(x_i|\text{Parents}(x_i))P(y|\text{Parents}(x_i)).$$

Гипотеза о факторизации – это предположение о том, что совместная вероятность есть произведение условных вероятностей каждого узла при известных значениях родителей:

$$P(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} P(x_i|\text{Parents}(X_i)).$$

Гипотеза о разделении представляет собой предположение о том, что для множеств вершин X, Y, Z, $X \cap Y = \oslash$, $X \cap Z = \oslash$, $Z \cap Y = \oslash$ справедливо утверждение о том, что если Z разделяет множества X, Y, то X, Y являются условно независимыми при известных значениях Z. Говорят, что Z разделяет множества X, Y, если оно разделяет все пары вершин, входящие в X и Y, блокируя все маршруты между соответствующими переменными.

Для определения факта условной независимости определенной вершины от всех остальных узлов сети необходимо определить ее марковское покрытие (МП). Под марковским покрытием понимается некоторое множество, в которое входит сама вершина, ее родительские, дочерние вершины, а также множество родителей дочерних вершин. Марковское покрытие для переменной $y \in X$ принято обозначать M^y.

Для динамической байесовской сети обозначим через X_t множество вершин слоя t, а через E_t – множество свидетельств (наблюдаемых переменных) слоя t. Полное совместное распределение вероятностей для ДБС [4] с учетом связей между слоями и свидетельствами на каждом слое можно записать в виде:

$$P(X_0, X_1, \ldots, X_t, E_1, \ldots, E_t) = P(X_0)\prod_{i=1}^{t} P(X_i \mid X_{i-1})P(E_i \mid X_i), \tag{1}$$

где $P(X_0)$ – начальное распределение вероятностей, $P(X_i \mid X_{i-1})$ и $P(E_i \mid X_i)$ – соответственно модели перехода и восприятия.

Выражение (1) выполняется в рамках ограничений, связанных с правилом формирования вероятностных связей при построении модели перехода. Предполагается, что рассматривается марковский процесс I рода и, как следствие, учитываются связи лишь в двух смежных временных срезах.

Рассмотрим используемые в работе гибридные алгоритмы обучения структуры и параметров ДБС. Обучение структуры и параметров ДБС тесно связано с понятиями условной независимости и МП. Обучение структуры применяется для формирования оптимальной топологии сети, учитывающей специфику процессов тестирования веб-приложений.

Для построения структуры статических байесовских сетей широко используется алгоритм минимаксного восхождения [5], который позволяет оптимизировать процедуру построения структуры байесовской сети за счет разделения логики построения базовой топологии сети и определения ее направленности. В качестве оценочной функции используется функция правдоподобия. Проблемные позиции данного алгоритма связаны с тем, что в результате его применения даже для статических байесовских сетей может быть получен локальный оптимум, а также необходимо расширение его функциональности для нахождения структуры динамических байесовских сетей. При применении алгоритма минимаксного восхождения к динамическим байесовским сетям его нужно адаптировать к построению первичной структуры байесовской сети для начального момента времени, и структуры связей для моделей перехода и восприятия. В рамках данного исследования предполагается разработка гибридного алгоритма, позволяющего сочетать статистические подходы к оценке топологии байесовских сетей и алгоритмы определения направленности связей между отдельными узлами байесовских сетей. Для определения топологии предполагается использование алгоритма на основе вычисления марковского покрытия, формирующего модель начального состояния и модель перехода между состояниями.

Методика предлагаемого алгоритма основывается на эвристическом анализе возможных связей, а именно определение множества родительских и дочерних вершин для некоторой вершины z на основе вычисления марковского покрытия M'. В основе вычисления марковского покрытия лежит процесс выполнения тестов на условную независимость. Использование данных тестов обусловлено необходимостью оценки устойчивости связей между дочерними и родительскими узлами. В основе вычисления тестов на условную независимость вершин x_i и x_j при наличии x_k в предложенном алгоритме используется G^2 критерий:

$$G^2 = 2\sum_{a,b,c} N_{i,j,k}^{a,b,c} \ln \frac{N_{i,j,k}^{a,b,c}}{E_{i,j,k}^{a,b,c}} = 2\sum_{a,b,c} N_{i,j,k}^{a,b,c} \ln \frac{N_{i,j,k}^{a,b,c} N_3^c}{N_{i,k}^{a,c} N_{j,k}^{b,c}}, \quad E_{i,j,k}^{a,b,c} = \frac{N_{i,k}^{a,c} N_{j,k}^{b,c}}{N_k^c}, \tag{2}$$

с числом степеней свободы:

$$df = (|D_m(x_i)| - 1)(|D_m(x_j)| - 1)\prod_{x_k} |D_m(x_k)|, \tag{3}$$

где $E_{i,j,k}^{a,b,c}$ и $N_{i,j,k}^{a,b,c}$ – количество всех ожидаемых при выполнении гипотезы об условной независимости и наблюдаемых в обучающей выборке D частот событий, заключающихся в том, что

$x_i = a$, $x_j = b$, $x_k = c$; $D_m(x_k)$ – множество возможных значений (домен значений), которые может принимать узел ДБС x_k.

G-критерий в рамках реализации процедуры обучения структуры ДБС выступает в качестве оценки связи между узлами, позволяя сформировать ненаправленную структуру ДБС. Для определения направленности связей в ДБС могут использоваться различные оптимизационные алгоритмы. Процедура поиска представляет собой рекурсивную операцию по добавлению, удалению или изменению направленности ребер графа, что в свою очередь приводит к изменению параметров графа G. В качестве целевой функции используются статистические критерии на основе логарифма правдоподобия, в частности, критерии Шварца и Акаике. Обобщенное математическое представление данных критериев имеет следующий вид:

$$\begin{gathered}\Phi(M) = L(G, X, D) - mF(N),\\ L(G, X, D) = \log P(D|X) = \sum_{i=1}^{n} \log P(D_i|X),\end{gathered} \tag{4}$$

где m – общее число параметров ДБС, D – обучающая выборка, G – статическая БС, X – переменные, входящие в состав ДБС.

Из обобщенного равенства (4) можно получить выражения для критерия Шварца и Акаике, определяя значение $F(N)$ соответственно равным $F(N) = \log N/2$ и $F(N) = 1$. Задачи локального поиска решаются по отношению к каждому узлу ДБС с целью определения максимума(минимума) оценочной функции, формируемой на основе данных критериев. Одним из достаточно эффективных алгоритмов поиска экстремумов является алгоритм Левенберга–Марквардта (ЛМ, являющегося разновидностью класса регуляризованных алгоритмов на основе метода Гаусса–Ньютона (ГН)). Основным отличием подхода на основе ЛМ от классического ГН является вычисление приближенного значения матрицы Гессе H' с учетом информации относительно вторых производных. Входными данными алгоритма ЛМ является выборка $D = \{(Z_k, Y_k)\}_{k=1}^{n}$, $y_k \in \mathbb{R}^m$ – вектор ожидаемых значений, а также регрессионная модель, задаваемая в виде функции $f(Q, Z_k)$, $Q = (q_1, \ldots, q_m)$ – вектор параметров модели. В результате целевую функцию можно записать в виде

$$E_D(Q) = \frac{1}{2}\sum_{k=1}^{n} [e_k(Q)]^2, \tag{5}$$

где $e_k(Q) = y_k(Q) - f(Q, Z_k)$.

Приближенное значение матрицы Гессе для рассматриваемой задачи будет иметь следующий вид:

$$\begin{gathered}H'(Q) = [J(Q)]^m J(Q) + R(Q),\\ J(Q) = \begin{pmatrix} \dfrac{de_1}{dq_1} & \cdots & \dfrac{de_1}{dq_m} \\ \cdots & \cdots & \cdots \\ \dfrac{de_n}{dq_1} & \cdots & \dfrac{de_n}{dq_m} \end{pmatrix},\end{gathered} \tag{6}$$

где $R(Q)$ – компоненты вторых производных, $J(Q)$ – матрица Якоби, $e(Q) = [e_1, e_2, \ldots, e_n]$ – функция отображения (функция невязки).

В основе алгоритма ЛМ лежит решение системы уравнений относительно приращения градиента ΔQ путем введения дополнительного коэффициента регуляризации λ и аппроксимации компоненты $R(Q)$

$$([J(Q)]^{\mathrm{T}} J(Q) + \lambda I(Q))\Delta Q = -[J(Q)]^{\mathrm{T}} E(Q), \tag{7}$$

где $E(Q) = y - f(Q)$ есть функция отображения (невязки).

В большинстве случаев целесообразно осуществить замену единичной матрицы $I(Q)$ на диагональную приближенную матрицу Гессе $\mathrm{diag}[H'(Q)]$. Это обусловлено тем, что снижение ско-

рости аппроксимации алгоритма ЛМ приводит к увеличению параметра λ. Как следствие, слагаемое $\lambda I(Q)$ теряет свой математический смысл:

$$([J(Q)]^{\mathrm{T}} J(Q) + \lambda \operatorname{diag}[H'(Q)])\Delta Q = -[J(Q)]^{\mathrm{T}} E(Q). \tag{8}$$

Применение алгоритма ЛМ накладывает ряд ресурсных и временных ограничений, связанных с необходимостью перерасчета матрицы Якоби $J(Q)$ на каждом шаге алгоритма. Основное время выполнения алгоритма ЛМ занимает расчет матрицы Якоби для каждой из выполняемых итераций. Для этого в рамках реализации гибридного алгоритма обучения ДБС наиболее оптимальным решением является комбинирование алгоритма ЛМ с методов секущих Бройдена. Метод Бройдена позволяет получить приближенную матрицу Якоби J_{k+1}. Определим соотношение секущих для получения выражения Бройдена

$$J_{k+1}(x_{k+1} - x_k) = F(x_{k+1}) - F(x_k). \tag{9}$$

Введем следующие обозначения: $\alpha = x_{k+1} - x_k$ и $\beta = F(x_{k+1}) - F(x_k)$. Далее определим формулу Бройдена

$$J_{k+1} = J_k + \frac{(\beta - J_k)}{\alpha\alpha^{\mathrm{T}}}\alpha^m. \tag{10}$$

Из выражения (10) следует, что необходимо получить точное значение якобиана J_0 на начальном этапе алгоритма, на последующих этапах формируется лишь приближенное значение якобиана J_{k+1}.

Для оценки транзитивных связей между временными состояниями динамической байесовской сети используется теория марковских цепей. Предполагается, что процесс перехода между данными состояниями представляет собой марковский процесс I рода. Структура выполнения алгоритма разделяется на два шага. Первый шаг характеризуется вычислением множества узлов-кандидатов, которые предположительно могут входить в марковское покрытие M' для текущей переменной z. На следующем шаге происходит удаление вершин, ошибочно добавленных в M'. Это достигается за счет повторного выполнения тестов на условную независимость для каждой переменной z при наличии всех возможных подмножеств M множества M'. Марковское покрытие для переменной z динамической байесовской сети определяется в виде

$$M_{t:t+1} = M_t \cup C_{t+1},$$

где C_{t+1} – множество дочерних вершин переменной z из временного среза $t+1$.

Общая структура алгоритма построения ненаправленной динамической байесовской сети состоит из следующих шагов.

Шаг 1. Задаются начальные значения для выполнения алгоритма: текущая переменная z, множество обучающих данных D, множество кандидатов $M'_{t:t+1} = \varnothing$.

Шаг 2. Выполняются итерации среди всех узлов сети и поиск вершин f с максимальной значимостью устойчивости связи между целевой переменной z и текущей переменной x_i, при анализе всех возможных подмножеств $M^* \subset M'_{t:t+1}$. После чего f добавляется в множество $M'_{t:t+1}$.

Шаг 3. Происходит формирование результирующего марковского покрытия для каждой из вершин путем удаления узлов ошибочно добавленных в $M'_{t:t+1}$, которые определяются за счет проведения G^2 тестов на условную независимость при наличии подмножеств $M^* \subset M'_{t:t+1}$. Если соблюдается гипотеза о разделении, то текущая вершина удаляется из $M'_{t:t+1}$.

В результате выполнения алгоритма мы получаем марковское покрытие для каждого узла динамической байесовской сети, на основе которого строится временная модель для каждого из узлов ДБС. Такой подход позволяет учесть наличие связей между срезами ДБС, а также определить, какие именно вершины имеют связи в соседних временных срезах. Построение ненаправленной структуры динамической байесовской сети не позволяет определить направление связей внутри сети, следовательно, возникает необходимость использования алгоритмов локального поиска. Применительно к семантике динамической байесовской сети в качестве функции оценки используется критерий Шварца или Акаике, напрямую зависящих от добавления, удаления и изменения направленности связей между узлами байесовской сети.

В качестве алгоритма локального поиска, предназначенного для определения направленности связей между узлами сети, предлагается использовать модификацию градиентного вывода, в частности алгоритм Левенберга–Марквардта. Данный алгоритма сочетает в себе алгоритм градиентного поиска и метод Гаусса–Ньютона. Основным отличием подхода на основе Левенберга–Марквардта от классического Гаусса–Ньютона является возможность вычисления приближенного значения матрицы Гессе. Применение алгоритма Левенберга–Марквардта (ЛМ) накладывает ряд ресурсных и временных ограничений, связанных с необходимостью перерасчета матрицы Якоби на каждом шаге алгоритма. Основное время выполнения алгоритма ЛМ занимает расчет матрицы Якоби для каждой из выполняемых итераций. Для преодоления ограничений классического алгоритма ЛМ в рамках реализации гибридного алгоритма обучения ДБС наиболее оптимальным решением является комбинирование алгоритма Левенберга–Марквардта и метода секущих Бройдена. Метод Бройдена позволяет получить приближенную матрицу Якоби.

Можно определить два основных этапа гибридного алгоритма обучения структуры ДБС с использованием алгоритма Левенберга–Марквардта и метода Бройдена. На первом этапе происходит формирование множества узлов-кандидатов в состав МП для каждой из вершин путем выполнения тестов на условную независимость с использованием G^2-критерия. В результате получаем вершины и соответствующие им множества родительских и дочерних вершин для каждого из n временных срезов. На следующем этапе определяем направленности связей между узлами ДБС, применяя алгоритм Левенберга–Марквардта с методом Бройдена. Стоит отметить, что направленность связей между узлами, участвующими в формировании транзитивных связей, имеет смысл только в прямом направлении от среза $t + k$ к срезу $t + k + 1$. Если в результате выполнения алгоритма ЛМ возникают обратные по направлению связи, они должны быть исключены, так как переходный процесс между временными срезами является Марковским, исключающий формирование обратных связей.

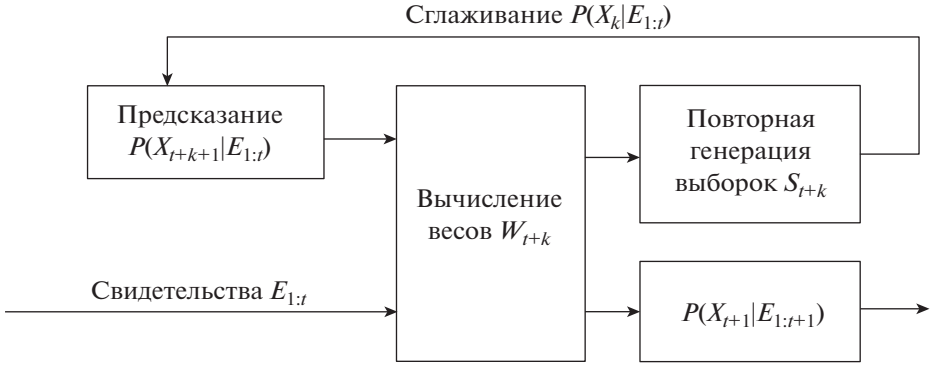
3. ГИБРИДНЫЕ АЛГОРИТМЫ ВЕРОЯТНОСТНОГО ВЫВОДА ДИНАМИЧЕСКИХ БАЙЕСОВСКИХ СЕТЕЙ

Вероятностный вывод является целевым инструментом любой стохастической модели. Применение вероятностного вывода в первую очередь направлено на получение апостериорного распределения при поступлении непрерывного потока свидетельств вплоть до текущего состояния системы. Основной реализацией вероятностного вывода во временных моделях является решение трех основных задач: фильтрация, предсказания и сглаживания. Наиболее эффективными алгоритмами вероятного вывода являются алгоритмы на основе метода Монте-Карло с применением цепей Маркова (МКМЦ), в частности, алгоритмы взвешивания с учетом правдоподобия (ВСП) и многочастичный фильтр (МЧФ).

В алгоритме ВСП осуществляется генерация только тех выборок S для переменных $X = (x_1, x_2, \ldots, x_n)$, которые в полной мере согласуются со свидетельствами E. Такое условие достигается за счет того, что в процессе выполнения вероятностного вывода происходит определение и фиксирование переменных свидетельств E, а формирование выборок осуществляется исключительно для всех оставшихся переменных, запроса Z и состояния X ДБС. В процессе выполнения алгоритма происходит формирование выборок для каждой из переменных состояния развернутой динамической байесовской сети, которые взвешиваются с учетом правдоподобия в соответствии с наблюдаемыми свидетельствами. Исходя из того, что любая переменная z зависит лишь только от родительских вершин, то распределение вероятностей для выборок может быть записано в виде следующего выражения:

$$S_w(Z, E) = \sum_{i=1}^{m} P(z_i \,|\, \mathrm{Parents}(z_i)), \quad E \subset \mathrm{Parents}(z_i). \tag{11}$$

Из формулы (11) следует, что в отличие от первичного распределения вероятностей $P(X)$ по всем переменным X, свидетельство E может вносить вклад в формирования вероятностного распределения выборок S_{ws}, так как E может входить в состав родительских вершин $\mathrm{Parents}(z_i)$. Весовая величина правдоподобия $w(z, E)$ определяется как разница между полученными и ожидаемыми распределениями вероятностей, сформированных для каждой переменной z выборок. Вес $W_{ws}(Z \,|\, E)$ вычисляется для каждой выборки Z и представляет произведение показателей



Фиг. 2. Обобщенная схема МЧФ фильтра.

правдоподобия свидетельств $E_i \in E$, если определено множество родительских вершин для каждой переменной свидетельства [6]:

$$W_{ws}(Z|E) = \sum_{i=1}^{l} P(E_i|\text{Parents}(E_i)). \tag{12}$$

Перемножая выражения (11) и (12), можно получить искомое соотношение для алгоритма взвешивания с учетом правдоподобия

$$S_{ws}(Z,E)W_{ws}(Z|E) = \sum_{i=1}^{m} P(Z_i|\text{Parents}(Z_i))\sum_{i=1}^{l} P(E_i|\text{Parents}(E_i)) = P(Z|E). \tag{13}$$

Классический алгоритм ВСП позволяет получить апостериорное распределение $P(X|E)$ за счет вычисления весов правдоподобия, однако с ростом общего числа переменных запроса и свидетельств, входящих в ДБС, наблюдаем увеличение доли выборок с низкими весами. Применение методов МКМЦ позволяет избежать данных проблем. Каждая новая генерация выборки S_{k+1} формируется на основе внесения случайного изменения в выборку S_k, полученную на предыдущем этапе выполнения метода МКМЦ. Одним из наиболее распространенных методов стохастического вероятностного вывода на основе метода МКМЦ является МЧФ фильтр [7]. Основным преимуществом МЧФ относительно других методов на основе МКМЦ является возможность использования различных подходов для оценки весов выборок, в частности рассмотренного нами метода ВСП. Такой подход позволяет комбинировать случайную генерацию методом Монте-Карло и оценку весов на основе подхода ВСП. В общем виде структура МЧФ фильтра приведена на фиг. 2.

Применительно к ДБС, выполнение МЧФ фильтрации осуществляется с учетом стохастических связей между узлами сети. Для этого поэтапно используются: начальное распределение $P(X_0)$, модель перехода $P(X_{t+1}|X_t)$ и модель восприятия $P(E_{t+1}|X_{t+1})$. Формирование начальной выборки S_t при наличии свидетельств $E_{1:t}$, полученных до текущего состояния, выполняется на основе распределения $P(X_t|E_{1:t})$. Формирование выборок S_{t+1} осуществляется за счет тиражирования свидетельств до момента времени $t+1$. Распределение вероятностей по всем выборкам для моментов времени t и $t+1$ определяется следующим образом [8]:

$$\begin{aligned} N'(X_t|E_{1:t}) &= N \times P(X_t|E_{1:t}), \\ N'(X_{t+1}|E_{1:t}) &= \sum_{X_t} P(X_{t+1}|X_t)N'(X_t|E_{1:t}). \end{aligned} \tag{14}$$

Процедура получения весов каждой из выборок основывается на применении алгоритма ВСП:

$$W(X_{t+1}|E_{1:t+1}) = P(E_{t+1}|X_{t+1})N'(X_{t+1}|E_{1:t}). \tag{15}$$

В результате получаем искомое распределение по всем N выборкам для момента времени $t+1$

$$N'(X_{t+1}|E_{1:t+1}) = N \times P(E_{t+1}|X_{t+1})N'(X_{t+1}|E_{1:t}) = N \times P(E_{t+1}|X_{t+1})\sum_{X_t} P(X_{t+1}|X_t)N'(X_t|E_{1:t}) =$$
$$= N \times P(E_{t+1}|X_{t+1})\sum_{X_t} P(X_{t+1}|X_t)P(X_t|E_{1:t}) = N \times P(X_{t+1}|E_{1:t+1}). \quad (16)$$

Распределение $N'(X_{t+1}|E_{1:t+1})$, представленное в формуле (14), зависит от общего числа выборок, используемых в процессе выполнения алгоритма МЧФ. В идеальном случае для получения требуемой точности алгоритма значение N должно выбираться достаточно большим $N \to \infty$, что накладывает ресурсные и временные ограничения на МЧФ алгоритм. Для оптимизации алгоритма МЧФ и снижения общего числа выборок, необходимых для достижения заданного уровня точности, предлагается использовать теорему Рао–Блеквелла–Колмогорова (РБК).

Сформулируем основные понятия достаточных статистик и теорему РБК. Под достаточной статистикой $T(X)$ будем понимать статистику относительно параметра θ, для которой условное распределение выборки $P(X|T(X))$ не будет зависеть от θ [9].

Если $T(X)$ является достаточной статистикой выборки X, а $T_1(X)$ – некоторая оценка параметра θ, тогда можно определить оценку $T_2(X) = \mathbb{E}_\theta(T_1(X)|T(X))$, для которой справедлива теорема РБК [10]

$$\mathbb{E}_\theta(T_1(X) - T_2(X))^2 \geq 0. \quad (17)$$

Исходя из формулы (11), можно установить соотношение дисперсий для оценок $T_1(X)$ и $T_2(X)$:

$$\mathbb{D}(T_1(X)) = \mathbb{E}(\mathbb{D}(T_1(X)|T(X))) + \mathbb{D}(\mathbb{E}(T_1(X)|T(X))) = \mathbb{E}(\mathbb{D}(T_1(X)|T(X))) + \mathbb{D}(T_2(X)),$$
$$\mathbb{D}(T_2(X)) \leq \mathbb{D}(T_1(X)). \quad (18)$$

Применение теоремы РБК для оптимизации МЧФ заключается в разделении множества переменных запроса X_t на подмножества $X_t' \subset X_t$ и $X_t'' \subset X_t$. В таком случае модель перехода будет иметь следующее представление:

$$P(X_{t+1}|X_t) = P(X_{t+1}'|X_{t:t+1}'', X_t')P(X_{t+1}''|X_t''). \quad (19)$$

В работах Дуста и Рассела [11] предполагается, что компонента $P(X_{t+1}'|E_{1:t-1}, X_{t+1}'')$ может быть определена аналитически еще до начала выполнения алгоритма МЧФ. Неизвестным остается лишь модель $P(X_{t+1}'|E_{1:t+1})$, которую и необходимо вычислить в процессе выполнения этапов МЧФ фильтра. В работе исследовано, что в большинстве случаев, применительно к семантике ДБС, компоненту $P(X_{t+1}'|E_{1:t-1}, X_{t+1}'')$ можно не задавать аналитически, а вычислить также в процессе выполнения фильтрации с помощью МЧФ. В таком случае апостериорное распределение вероятностей для следующего момента времени $t+1$, соответствующее переменным $X_t' \subset X_t$ и $X_t'' \subset X_t$, можно определить на основе цепного правила:

$$P(X_{t+1}', X_{t+1}''|E_{1:t+1}) = P(X_{t+1}''|E_{1:t+1})P(X_{t+1}'|X_{t:t+1}'', X_t')P(X_{t+1}''|X_t''). \quad (20)$$

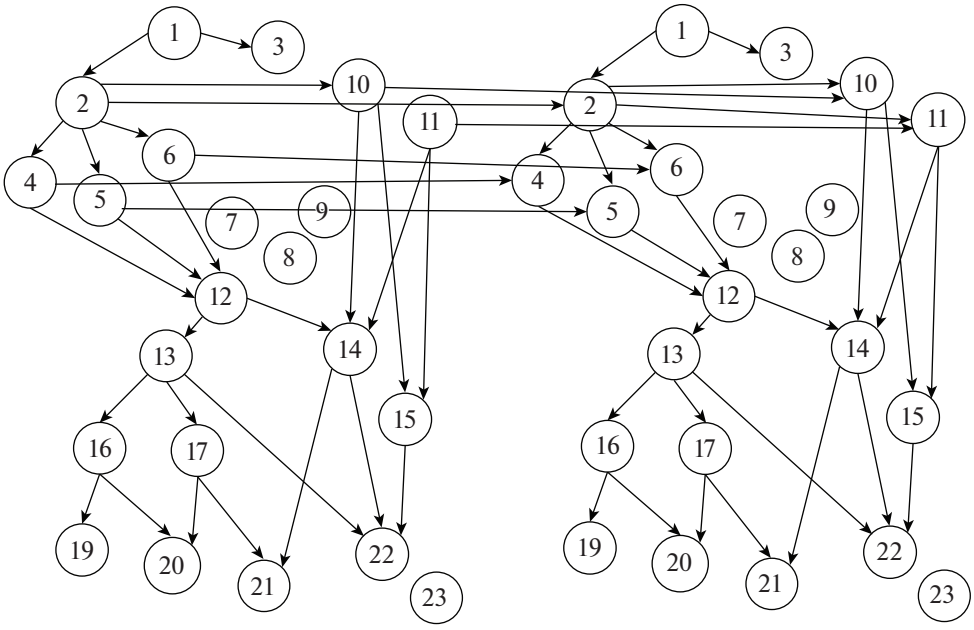
Применительно к алгоритмам ВСП, используемым в процессе определения весов выборок в МЧФ, можно определить оценку этих весов в терминах введенных обозначений:

$$W(X_{t+1}', X_{t+1}''|E_{1:t+1}) = P(E_{t+1}|X_{t+1}'', X_{t+1}')N'(X_{t+1}'', X_{t+1}'|E_{1:t}). \quad (21)$$

С учетом выражения (18) можно определить условие существования выборок для распределений $P(X_{t+1}'|E_{1:t+1})$ и $P(X_{t+1}', X_{t+1}''|E_{1:t+1})$ на основе использования теоремы РБК

$$\mathbb{D}(W(X_{t+1}'|E_{1:t+1})) \leq \mathbb{D}(W(X_{t+1}', X_{t+1}''|E_{1:t+1})). \quad (22)$$

Из неравенства (15) следует, что из апостериорного распределения вероятностей будут исключаться выборки с низким значением среднеквадратического отклонения, это приводит к тому, что вклад будут вносить лишь согласованные выборки. Для получения искомого апостериорного распределения $P(X_{t+1}'|E_{1:t+1})$ для момента времени $t+1$ введем ограничение, связанное с определением транзитивных связей между смежными временными срезами. Транзитивные свя-



Фиг. 3. Обученная структура ДБС фаззинга программных ошибок типа “инъекция”.

зи имеют смысл только между одними и теми же узлами, разнесенными между срезами. Определим распределение $P(X'_{t+1}|E_{1:t+1})$ за счет преобразования выражения (14):

$$P(X'_{t+1}|E_{1:t+1}) = N'(X''_{t+1}, X'_{t+1}|E_{1:t+1})/N. \quad (23)$$

4. ВЫЧИСЛИТЕЛЬНЫЙ ЭКСПЕРИМЕНТ ПО ПОСТРОЕНИЮ ДИНАМИЧЕСКОЙ БАЙЕСОВСКОЙ СЕТИ ФАЗЗИНГА ИНЪЕКЦИЙ, РЕШЕНИЕ ЗАДАЧ ВЕРОЯТНОСТНОГО ВЫВОДА

В рамках проведения вычислительного эксперимента рассмотрим тестирование систем управления сайтами (СУС). Система управления сайта представляет собой разновидность веб-приложения, предоставляющая расширенный набор по управлению, размещению и хранению различного содержимого. СУС используется как фреймворк для создания интерактивных веб-приложений с целым набором функциональных возможностей, включая механизмы обеспечения безопасности и валидации данных. Среди СУС лидирующую роль занимает WordPress, в рамках исследования остановимся на данном решении. Для обучения ДБС и получения обучающей выборки в качестве целевых приложений используется набор виртуальных машин с развернутыми СУС типа WordPress, начиная с версии 4.3 до 5.2. Система фаззинга строится в виде блочной архитектуры, где происходит автоматическая настройка среды тестирования определенного типа веб-приложений за счет классификации типовых структур приложений тестирования методом черного ящика и формирования списка входных компонентов (параметров), для которых необходимо провести тестирование. Формирование тестовых выборок происходит по результатам выполнения каждого из блоков фаззинга, являющихся узлами ДБС.

Полученная структура ДБС фаззинга “инъекций”, сформированная по результатам тестирования СУС WordPress и обучения на основе алгоритма ЛМ и метода Бройдена, приведена на фиг. 3. Описание узлов ДБС, являющихся блоками фаззинга, представлено в табл. 1.

Фигура 3 показывает, что на основе построенной ДБС формируется направленная система фаззинга. Направления указывают не только на условные зависимости вершин, но и устанавливают последовательность выполнения различных блоков фаззинга, а также взаимосвязь данных блоков. Узлы, не имеющие связей в процессе тестирования, будут отброшены. В построенной ДБС полная условная независимость допустима лишь для корневого узла сети. Необходимо отметить, что если условные связи имеют нулевое значение вероятности в ТУВ, то на следующих этапах тестирования такие блоки фаззинга будут пропущены ввиду их неэффективности к обнаружению ошибок типа “инъекции” для данного типа веб-приложений.

Таблица 1. Характеристика узлов динамической байесовской сети фаззинга инъекций веб-приложений

Узел	Характеристика
1	Определение типа инъекции: SQL, команд, кода
2, 3	Механизмы кодирования обхода межсетевых экранов веб-приложений (WAF)
4, 5, 6, 7, 8, 9	Типы инъекций: Time Based blind, Boolean Based Blind, Error Based Blind, Out of Band, Union injection, Stacked Time
10, 11	Инъекции команд и кода
12	Определение типа и версии СУБД, установленной на сервере
13	Получение структуры таблиц и баз данных СУБД
14	Исполнение команд операционной системы через SQL инъекции
15	Получение доступа к компонентам сети из командного интерфейса СУБД
15	Получения данных, хранящихся в таблицах базы данных
17	Возможность удаленной загрузки файлов, через функции СУБД
19, 20, 21, 22, 23	Нарушение механизмов аутентификации, авторизации, целостности, конфиденциальности и доступности

Таблица 2. Результаты сравнения времени выполнения различных алгоритмов обучения вывода в ДБС “инъекции”

№ п/п	Размер обучающей выборки D	Алгоритм ВС	Алгоритм АМП	Алгоритм ММВ	Алгоритм ЛМ Бройден
1	2000	0.63215 с	0.54231 с	0.49314 с	0.38201 с
2	50000	2.94313 с	2.16543 с	1.85324 с	1.45467 с
3	600000	10.57213 с	8.57732 с	7.02256 с	6.55224 с
4	1000000	18.18432 с	12.25452 с	11.05311 с	8.98432 с
5	10000000	40.09432 с	32.54146 с	28.19356 с	13.95421 с

В рамках проведения процедур обучения и вероятностного вывода используются распределенная вычислительная платформа Apache Hadoop YARN и распределенная файловая система HDFS. Данная платформа имеет в своем составе 6 вычислительных узлов, представленных серверами со следующей аппаратной конфигурацией: 2 процессора Intel Xeon-Platinum 2.5 GHz 16 ядер, 128 GB ОЗУ, жесткий диск 10 ТВ. Размер распределенной файловой системы HDFS 59.5 ТВ, канал связи между узлами обеспечивают скорость до 16 Gb в секунду. Распределенная файловая система Hadoop HDFS используется для хранения обучающих выборок, а также промежуточных данных: таблицы условных вероятностей, весовые распределения выборок, полученные методом МКМЦ. При этом часть данных, используемых в процессе выполнения гибридных алгоритмов обучения и вероятностного вывода, хранится непосредственно в оперативной в виде распределенного множества данных, представленного программной реализацией Apache Hadoop YARN. YARN представляет собой разновидность MapReduce с встроенным планировщиком нагрузки [12], распределения ресурсов и модулем отказоустойчивости, построенным по архитектуре клиент–сервер, с выраженным управляющим узлов (мастер–узел) и клиентскими узлами (рабочий узел). Из 6 узлов вычислительной системы один узел используется нами одновременно в качестве рабочего и мастер-узла. Это дает возможность задействовать все ресурсы 6 узлов в процессе решения задач обучения и вероятностного вывода. Далее в табл. 2 и 3 приведем результаты вычислительных экспериментов по оценке времени выполнения общеизвестных алгоритмов обучения (алгоритм возрастания-сокращения (ВС), инкрементных ассоциаций марковского покрытия (ИАМП), минимаксного восхождения (ММВ)) и вероятностного вывода (Метрополиса–Гастингса (МГ), выборки по значимости (ВЗ), МЧФ) и разработанных гибридных алгоритмов.

Подводя итоги вычислительного эксперимента, отметим, что разработанные алгоритмы обучения и вероятностного ДБС являются ресурсно-эффективными и достаточно легко масштабируются для выполнения на любой из параллельных систем. Отметим, что рассмотренные в ка-

Таблица 3. Результаты сравнения времени выполнения различных алгоритмов вероятностного вывода в ДБС “инъекции”

№ п/п	Размер выборки S	Алгоритм МГ	Алгоритм ВЗ	Алгоритм ВСП	Алгоритм МЧФ	Алгоритм МЧФ РБК
1	2000	0.12311 с	0.13456 с	0.23564 с	0.17654 с	0.10231 с
2	50000	4.33784 с	3.26546 с	3.31231 с	2.26766 с	2.05322 с
3	600000	8.65432 с	7.76532 с	7.81111 с	6.54355 с	5.44355 с
4	1000000	25.23121 с	26.42982 с	22.12941 с	20.31234 с	15.87431 с
5	10000000	56.26332 с	48.21942 с	36.98765 с	31.54328 с	21.12453 с

честве сравнения существующие алгоритмы обучения структуры ВС, ИАМП и ММВ имеют ряд существенных недостатков. Первый из них связан с тем, что они адаптированы для обучения лишь статических БС. Второй недостаток связан с использованием метода восхождения для определения направленностей связи между узлами ДБС. Недостаток заключается в том, что алгоритмы обладают достаточно большой вероятностью попадания оценочных функций в локальный оптимум. Из этого следует, что направленность между узлами ДБС будет задана некорректно, а полученная структура ДБС не может быть в полной мере использована для решения задач вероятностного вывода. Алгоритм ЛМ в сочетании с методом Бройдена позволяет повысить точность расчета экстремумов оценочных функций на основе логарифма правдоподобия, метрик Шварца и Акаике, а также исключить возможность получения некорректной структуры ДБС.

5. ЗАКЛЮЧЕНИЕ

Разработанные в работе алгоритмические и программные решения для оптимизации процедур фаззинга веб-приложений позволяют осуществлять тестирование, обучение и накопление статистических данных. Модели ДБС служат для представления внутренних процессов фаззинга и построения функциональных связей между отдельными блоками фаззинга. Применение такого подхода позволяет осуществлять настройку средств тестирования под специфику анализа определенных групп ошибок. При этом число срезов будет пропорционально количеству настроек или изменений, вносимых в веб-приложения в рамках расширения их функциональных возможностей или совершенствования механизмов защиты. Рассмотрена возможность применения теоремы РБК в рамках многочастичного фильтра, что позволяет оптимизировать МЧФ и гарантировать заданную точность, но при меньшем числе выборок.

СПИСОК ЛИТЕРАТУРЫ

1. *Zalewski M.* The Tangled Web. A Guide to Securing Modern Web Applications. San Francisco: No starch Press, 2012. P. 477.
2. *Саттон М., Амини П.* Fuzzing: исследование уязвимостей методом грубой силы / Пер. с англ. М.: Вильямс, 2009. С. 560.
3. *Тулупьев А.Л., Сироткин А.В., Николенко С.И.* Байесовские сети доверия: логико-вероятностный вывод в ациклических направленных графах. СПб.: Изд-во Санкт-Петербурского ун-та, 2009. С. 400.
4. *Korb A., Nicholson A.* Bayesian Artificial Intelligence. London: Chapman & Hal, CRC Press UK, 2004. P. 244.
5. *Tsamardinos I., Brown L.E., Aliferis C.E.* The max-min hill-climbing Bayesian network structure learning algorithm // Machine Learning. 2006. P. 31–78.
6. *Торопова А.В.* Подходы к диагностике согласованности данных в байесовских сетях доверия // Труды СПИИРАН. 2015. № 43. С. 156–178.
7. *Azarnova T.V., Polukhin P.V.* Advanced hybrid stochastic dynamic Bayesian network inference algorithm development in the context of the web applications test execution // IOP Conf. Ser. 2019. P. 052028–052035.
8. *Russel S., Norvig P.* Artificial Intelligence A Modern Approach. Boston: Prentice Hall, 2009. P. 1095.
9. *Кельберт М.Я., Сухов Ю.В.* Вероятность и статистика в примерах и задачах / Пер. с англ. Основные понятия теории вероятностей и математической статистики. М.: МЦНМО, 2007. С. 241–285.
10. *Колмогоров А.Н.* Теория вероятностей и математическая статистика. М.: Наука, 2005. С. 581.
11. *Deucet A., Freitas N., Murphy K.P., Russel S.* Rao-Blackwellised Particle Filtering for Dynamic Bayesian Networks // Proc. of 16th Conf. Uncertainty in AI. 2000. P. 176–183.
12. *Zaharia M., Chowdhury M., Das T., Dave A., McCauley M., Franklin M., Shenker S., Stoica I.* Resilent Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing // NSDI. 2012. P. 1–15.