Динамические Байесовские сети как инструмент тестирования веб-приложений методом фаззинга

Т. В. Азарнова, П.В. Полухин

Воронежский государственный университет

ММРО 2019

Москва ● 27 ноября 2019

Цели и задачи исследования

Цель исследования - заключается в разработке ***моделей, методов, алгоритмов и программного обеспечения*** для реализации процедур тестирования веб-приложений методом фаззинга, на основе динамических байесовских сетей

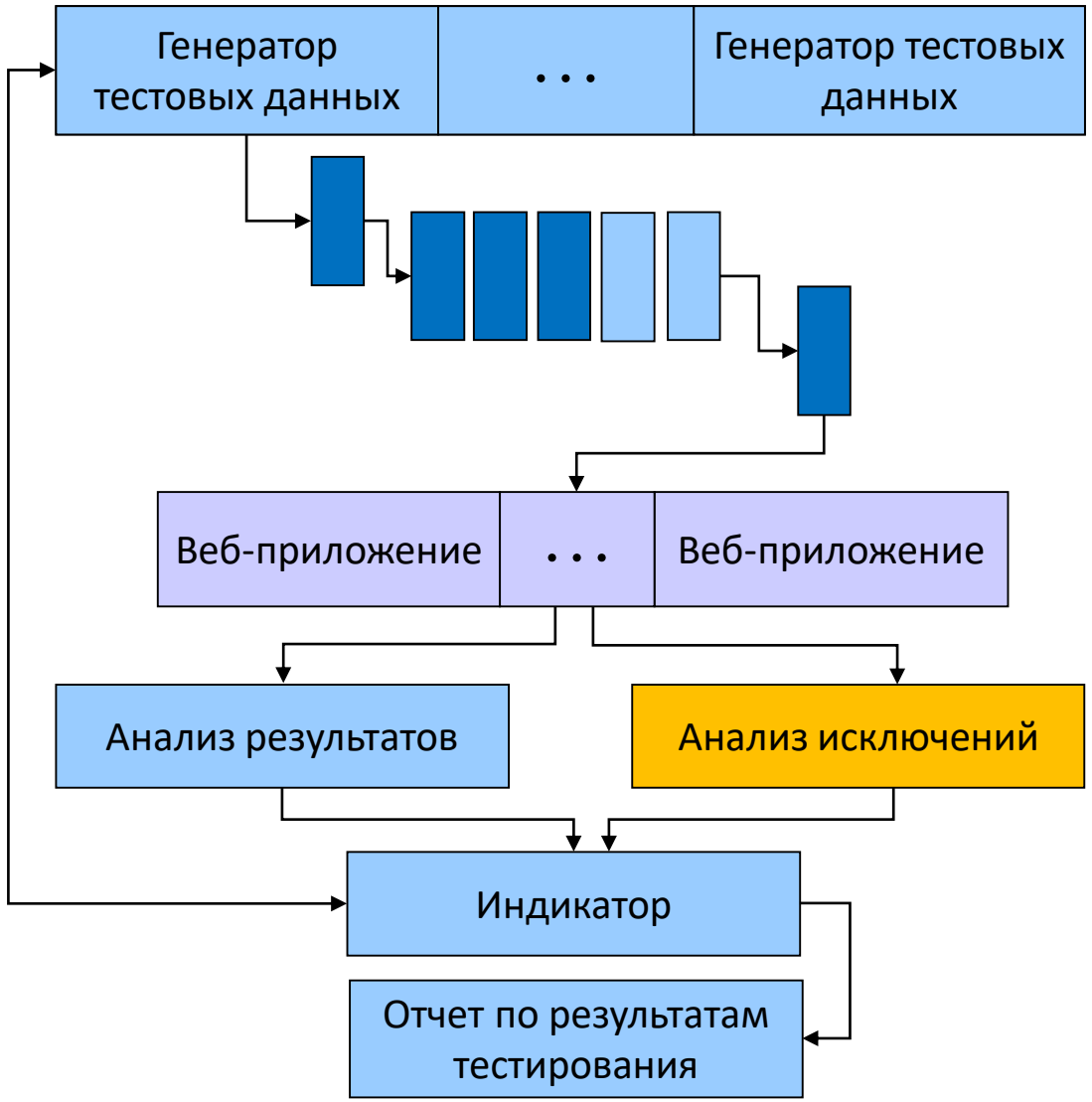
Задачи исследования

- Провести агрегированную структуризацию информации о тестировании веб-приложений методом фаззинга, объединяющую в единую концепцию: функциональную модель процесса тестирования, логико-вероятностную структуру информации и методологическую базу обнаружения ошибок.
- Построить статические и динамические байесовские модели для управления процессом тестирования методом фаззинга основных классов ошибок веб-приложений.
- Разработать адаптированные к процедурам фаззинга гибридные алгоритмы обучения предложенных динамических байесовских моделей.
- Разработать гибридные алгоритмы фильтрации, прогнозирования и сглаживания для решения задач ретроспективного анализа и прогнозирования в процедурах тестирования веб-приложений методом фаззинга.

Понятие фаззинга

Фаззинг – метод обнаружения ошибок в программном обеспечении, заключающийся в подаче на вход исследуемого объекта заведомо некорректных данных с целью вызова события сбоя или ошибки;

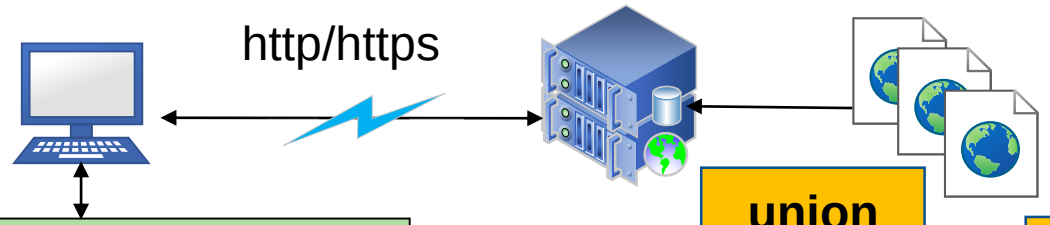
Генератор тестовых данных – механизм порождения наборов тестов, исходя из специфики обнаружения определенного типа программных ошибок



Классификация уязвимостей веб-приложений

A1.Инъекция кода	Инъекции SQL команд и кода могут возникнуть, когда непроверенные данные отправляются в приложение как часть команды или запроса. Сторонний пользователь может внедрить свой подзапрос или команду с целью получения данных без надлежащего на то разрешения.
A2.Некорректная аутентификация и управление сессиями	Механизмы аутентификации и управления сессиями могут быть реализованы неправильно, позволяя стороннему пользователю скомпрометировать пароли, сессии или использовать ошибки с целью получения идентификационных данных пользователей.
A3.Межсайтовый скриптинг	XSS ошибки возникают в ситуации, когда приложение получает данные из вне и выводит их без всякой фильтрации и экранирования. XSS открывают возможность исполнять скрипты внутри веб-браузера пользователя, тем самым позволяя получить сессии пользователей, изменить контент веб-страницы, перенаправлять пользователей на сторонние ресурсы.
A4.Небезопасные прямые ссылки на объекты	Прямые ссылки на объект возникают в тех случаях, когда разработки предоставляют доступ к внутренним объектам своего приложения: файлам и каталогам, которые могут содержать данные для доступа к СУБД или аутентификационные данные пользователей. Используя это, сторонник пользователь может получить доступ к критичным данным приложения.
A5.Небезопасная конфигурация	Хорошая устойчивость и безотказность требует грамотной конфигурации приложений, фреймворков, веб-серверов, серверов баз данных и исполняемой платформы. Критерии настройки должны быть определены, детально изучены и установлены, так как значения по умолчанию не соответствуют установленным требованиям.
A6.Утечка чувствительных данных	Сохранность данных требует дополнительные механизмы шифрования и другие меры предосторожности при обмене и хранении данных.
A7.Отсутствие контроля доступа к функциональному уровню	Большинство веб-приложений проверяют уровень доступа, прежде чем дать пользователю расширенные возможности. Тем не менее, в приложении должны производиться такие проверки для каждой из функций, реализующей расширенный функционал.
A8.Подделка межсайтовых запросов	CSRF позволяет модифицировать HTTP запросы из браузера некоторого пользователя, в том числе cookie и другие аутентификации доступные в сеансе пользователя.
A9.Использование компонентов с известными уязвимостями	Программные компоненты, такие как библиотеки классов, фреймворки, как правило, работают с полными системными привилегиями. Большинство программных компоненты содержат критичные ошибки, могут повлечь потерю данных, получение контроля над сервером.
A10.Небезопасные перенаправления	Веб-приложения часто перенаправляют пользователей на другие страницы и веб-сайты, однако используют нефильтрованные данные для определения целевого ресурса. Без соответствующей проверки можно перенаправить пользователя на сторонние ресурсы.

Типовой сценарий фаззинга «SQL- инъекций»



Получение списка пользователей
и паролей СУБД PostgreSQL v.12

https://my-site.com?id=20 + **'** + **union ;** + **select usename, passwd FROM pg_shadow--**

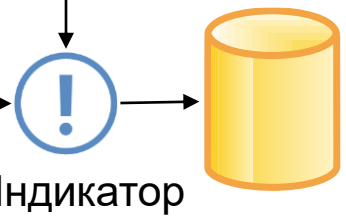
Веб-приложение
(пользовательский интерфейс)

```
<html>
 <body>
 <div class="body_padded">
 <div class="vulnerable_code_area">
  <h3>User ID:</h3>
  <form action="#" method="POST">
    <input type="text" name="id">
    <input type="submit" name="Submit" value="Submit">
  </form>
    <div>
       First name: admin<br>
       Surname: administrator<br>
       Email: dmurphy@gmail.com
    </div>
    <div>
      First name: 8hd32cg <br>
      Surname: postgres
      Email: md532e12f215ba27cb750c9e093ce4b5127
    </div>
 </body>
</html>
```

хэш пароля в формате md5

Веб-служба
(межпрограммное взаимодействие)

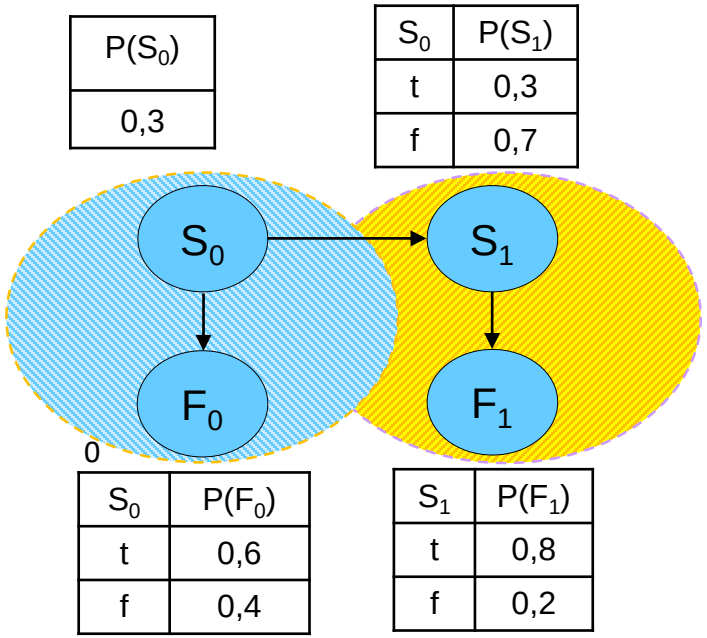
```
{
 "count" : "1",
 "group" : "users",
 "status" : "active"
 "timestamp" : "1574426602"
 "data": [
  {
    "name: admin",
    "surname": administrator",
    "email": "dmurphy@gmail.com"
  },
  {
    "name": " g8ol92ck ",
    "surname": "postgres",
    "email":
    "md532e12f215ba27cb750c9e093ce4b5127"
  }
 ]
}
```



Индикатор

$$D = \{(x_k, y_k)\}_{k=1}^{N}$$

Семантика динамических байесовских сетей (ДБС)



Марковский процесс первого порядка:
$P(X_t|X_{0:t-1}) = P(X_t|X_{t-1})$,
$P(E_t|X_{0:t}, E_{0:t-1}) = P(E_t|X_t)$

Полное совместное распределение вероятностей:

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

где X_t, E_t – множество ненаблюдаемых и наблюдаемых (свидетельств) переменных для момента времени , $P(X_0)$ – начальное распределение вероятностей, $P(X_i|X_{i-1})$ – модель перехода, $P(E_i|X_i)$ – модель восприятия

Алгоритм для построения байесовской сети:

Д. Перл

- Выбор множества переменных $\{X_i\}$, описывающих заданную предметную область;
- Определение порядка переменных $< X_1, \dots, X_n >$;
- Пока есть переменные слева:
  - добавляем следующую вершину X_i в сеть;
  - добавляем дуги из некоторого минимального множества узлов, которые уже находятся в сети из $Parents(X_i)$ в X_i, при этом учитывается свойство условной независимости $P(X_i|Parents(X_i), X_1', \dots, X_m') = P(X_i|Parents(X_i))$, где $X_1', \dots, X_m'$ – множество переменных предшествующих X_i, которые не входят в $Parents(X_i)$ $(X_1', \dots, X_m' \notin Parents(X_i))$;
- определяем таблицу условных вероятностей для X_i.

Критерии условной независимости вершин ДБС

Условная независимость вершин X и Y в ДБС

$$P(X,Y|P(X_i)) = P(X|P(X))P(Y|P(Y))$$

$$P(X) = \int_I p(X)dX = \sum_I p(X)$$

Критерий Пирсона

$$\chi^2(X,Y|Z) = \sum_{a,b,c} \frac{\left(N_{a,b,c} - E_{a,b,c}\right)}{E_{a,b,c}}$$

$$E_{a,b,c} = \frac{N_{ac}N_{bc}}{N_c}$$

$E_{a,b,c}$ – ожидаемое число выборок, $x = a, y = b, z = c, N_{a,b,c}$ – частота появления данных, где x=a, y=b, z=c

G-критерий

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

$$E_{i,j,k}^{a,b,c} = \frac{N_{i,k}^{a,c}N_{j,k}^{b,c}}{N_k^c}$$

$E_{i,j,k}^{a,b,c}$ и $N_{i,j,k}^{a,b,c}$ – множества всех ожидаемых и возможных повторений для обучающей выборки D, где $X_i = a, X_j = b, X_k = c$.

Алгоритм построения ненаправленной ДБС

- Задаются начальные значения для выполнения алгоритма: текущая переменная Z, множество обучающих данных D, множество кандидатов для марковского покрытия ${M_{t:t+1}}' = \emptyset$.
- Выполняется итерация среди всех узлов сети X и расчет максимального значения F среди минимальных значений, характеризующих устойчивость связи между целевой переменной Z и текущей переменной X_i, при наличии всех возможных подмножеств $M^* \subset {M_{t:t+1}}'$. После чего F добавляется во множество ${M_{t:t+1}}'$.
- На следующем шаге происходит формирование результирующего марковского покрытия для каждой из вершин за счет удаление узлов ошибочно добавленных в ${M_{t:t+1}}'$ за счет проведения G тестов на условную независимость вершины Z при наличии подмножеств $M^* \subset {M_{t:t+1}}'$. Если соблюдается правило d – разделенности, то текущая вершина удаляется из ${M_{t:t+1}}'$.

Методы поиска экстремума функции

- Поиск с восхождением
- Поиск с запретами
- Имитация отжига
- Метод Гаусса-Ньютона
- Метод Коши
- Метод Левенберга-Марквардта
- Метод Бройдена-Флетчера-Гольдфарба-Шанно
- Градиентный криволинейный поиск

Формирование оценочных функции для ДБС

Логарифм Правдоподобия

$$L(G, \theta^G, D) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{i,j,k} log \frac{N_{i,j,k}}{N_{i,j}}$$

Оценка максимального правдоподобия

$$\hat{G} = \underset{G}{\operatorname{argmax}} \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{i,j,k} log \frac{N_{i,j,k}}{N_{i,j}}$$

Метрика Шварца и Акаике

$$Q(M) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{i,j,k} log \frac{N_{i,j,k}}{N_{i,j}} - \sum_{i=1}^{r_i} (r_i - 1)\, q_i F(N)$$

$F(N) = 1$ *для критерия Акаике*

$F(N) = \log N/2$ *для критерия Шварца*

Метрика Байеса-Дирихле

$$ln\, P(D|G) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} ln \frac{\Gamma\left(\sum_{k=1}^{r_i} \alpha_{i,j,s}\right)}{\Gamma\left(\sum_{s=1}^{r_i} N_{i,j,s} + \alpha_{i,j,s}\right)} + \sum_{s=1}^{r_i} \frac{\Gamma\left(N_{i,j,s} + \alpha_{i,j,s}\right)}{\Gamma\left(\alpha_{i,j,s}\right)}$$

$$D = \{D_1 = \{X_1^G = 1, X_2^G = 2, \dots, X_3^G = n\}, D_2 = \{X_1^G = 2, X_2^G =$$

Эквивалентная метрика Байеса-Дирихле

$$ln\, P(D|G) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} ln \frac{\Gamma\left(\frac{1}{q_i}\right)}{\Gamma\left(\sum_{s=1}^{r_i} N_{i,j,s} + \frac{1}{q_i}\right)} + \sum_{s=1}^{r_i} \frac{\Gamma\left(N_{i,j,s} + \frac{1}{q_i r_i}\right)}{\Gamma\left(\frac{1}{q_i r_i}\right)}$$

$$\alpha_{i,j,k} = \alpha(x_i = k, P(x_i) = j) = \frac{1}{r_i q_i}$$

Расчет приближенной матрицы Гессе

$$H'(w) = [J(w)]^T J(w) + R(w)$$

$$J(w) = \begin{pmatrix} \frac{de_1}{dw_1} & \dots & \frac{de_1}{dw_n} \\ \dots & \dots & \dots \\ \frac{de_m}{dw_1} & \dots & \frac{de_m}{dw_n} \end{pmatrix}$$ – Якобиан

$e(w) = [e_1, e_2, \dots, e_n]^T$ – функция отображения (функция невязки для условия $m \geq n$)

$Q(w)$ – компоненты матрицы Гессе, содержащие значения производных высших порядков для w

Метод Левенберга

$$\left([J(w)]^T J(w) + \lambda I(w)\right)\Delta w = -[J(w)]^T E(w)$$

λ - параметр регуляризации Левенберга

Метод Левенберга-Марквардта

$$([J(w)]^T J(w) + \lambda diag[H'(w)])\Delta w = -[J(w)]^T E(w)$$

Соотношение секущих

$$a_k(x_{k-1}, x_k) = f(x_{k-1}) - f(x_k)$$

$$J_{k+1}(x_{k+1} - x_k) = F(x_{k+1}) - F(x_k)$$

Аффинная модель

$$\xi_{k-1} = F(x_{k-1}) + J_{k-1}(x - x_{k-1})$$

$$\xi_k = F(x_k) + J_k(x - x_k)$$

$$\xi_k(x) - \xi_{k-1}(x) = \alpha(J_k - J_{k-1})z + (J_k - J_{k-1})t$$

Метод Бройдена

$$J_{k+1} = J_k + \frac{(\beta - J_k)}{\alpha\alpha^T}\alpha^T$$

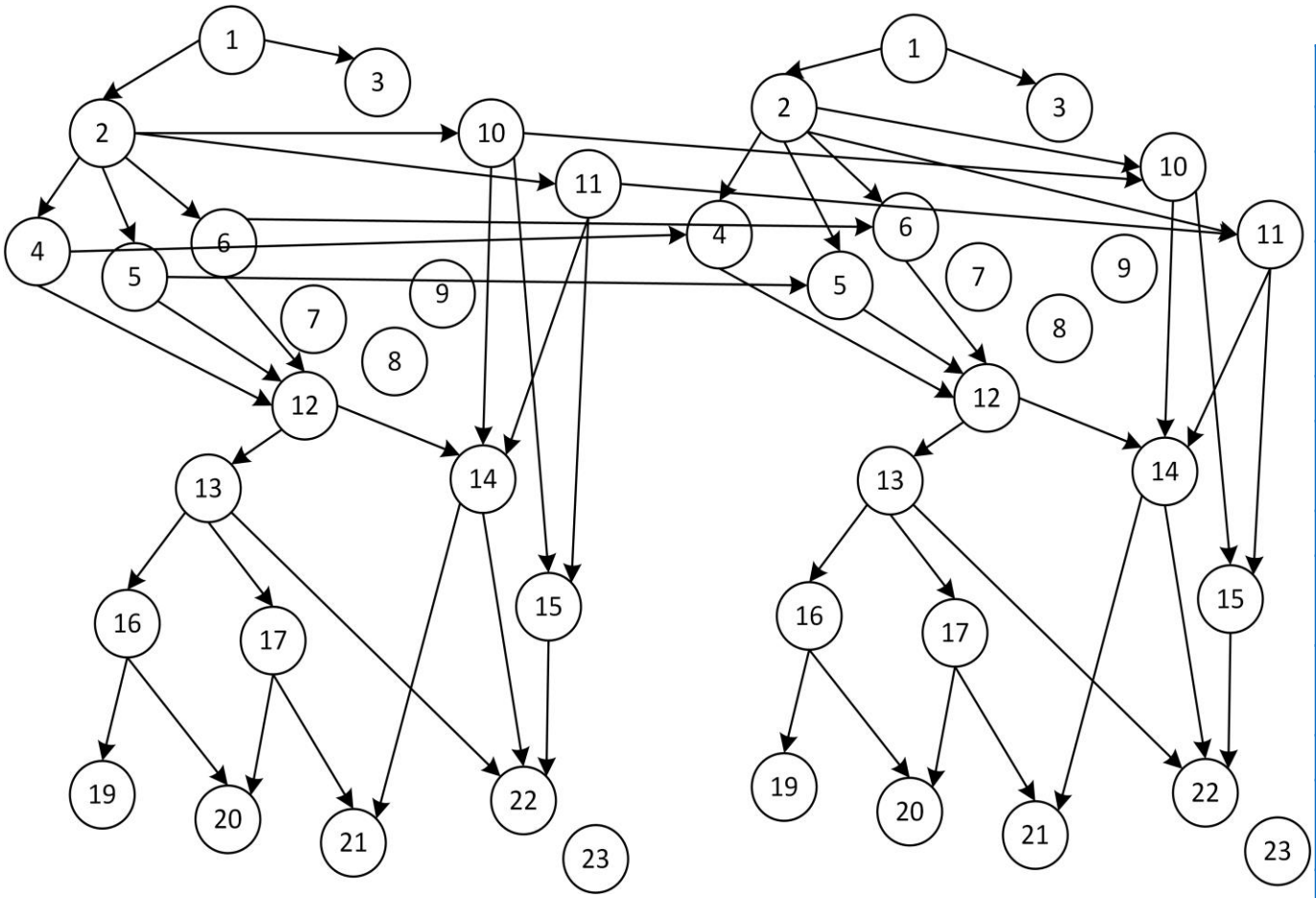
$$\alpha = x_{k+1} - x_k$$

$$\beta = F(x_{k+1}) - F(x_k)$$

Этапы гибридного алгоритма вычисления Марковского покрытия с применением цепей Маркова (МПМЦ) для обучения структуры ДБС

- На первом этапе происходит заполнение множеств $M_{t:t+1}$ на основе узлов-кандидатов в состав Марковского покрытия и исключения ошибочно добавленных переменных, за счет выполнения G-тестов.
- На втором этапе происходит определения направленности связей за счет вычисления оценок на основе алгоритма Левенберга-Марквардта

Обученная структура ДБС фаззинга «Инъекции»



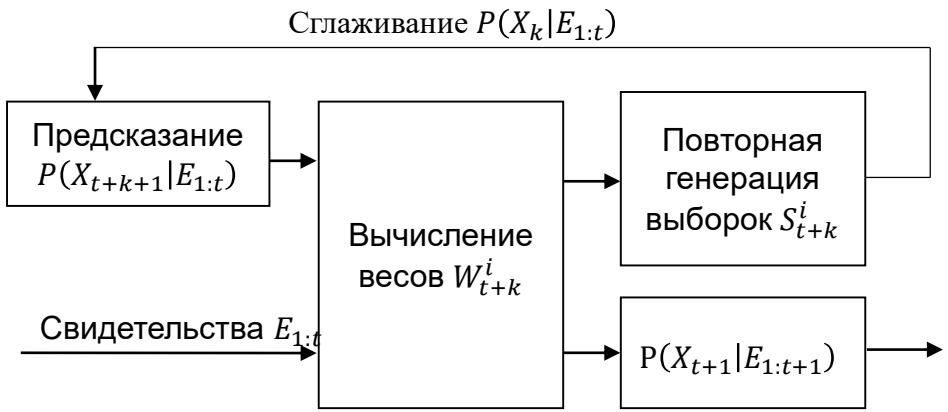
Характеристика узлов ДБС «Инъекции»

Узел	Характеристика
1	Определение типа инъекции: SQL, команд, кода
2,3	Механизмы кодирования и обхода межсетевых экранов веб-приложений (WAF)
4,5,6,7,8, 9	Различные 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 и команд внутри инъекции кода и 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

- Алгоритм Метрополиса-Гастингса
- Алгоритм Гиббса
- Алгоритм взвешивания с учетом правдоподобия
- Алгоритм формирования выборок по значимости
- Многочастичный фильтр

Алгоритм многочастичного фильтра (МЧФ)

Обобщенная схема МЧФ фильтра



Основная идея МЧФ фильтра

$$P(X_{t+1}|E_{1:t+1}) = \sum_{i=1}^{N_s} W_i^{t+1}\,\delta(X_{t+1}, X_{t+1}^i)$$

X_{t+1}^i – выборка, соответствующая переменной X_{t+1}

Распределение вероятностей по выборкам

$$N'(X_t|E_{1:t}) = N \times P(X_t|E_{1:t}),$$

$$N'(X_{t+1}|E_{1:t}) = \sum_{X_t} P(X_{t+1}|X_t)\,N'(X_t|E_{1:t})$$

Взвешивание весов с учетом правдоподобия

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

Апостериорное распределение вероятностей по всем выборкам

$$N'(X_{t+1}|E_{1:t+1}) = N \times P(E_{t+1}|X_{t+1})N'(X_{t+1}|E_{1:t})$$

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

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

$$= N \times P(X_{t+1}|E_{1:t+1})$$

Алгоритм МЧФ и теорема Рао-Блеквелла-Колмогорова

Достаточная статистика

$$T(X) = \big(T(X_1), T(X_2), \dots, T(X_n)\big)$$

Условное математическое ожидание в терминах достаточных статистик

$$\mathbb{E}_\theta(\mathrm{T}_2(\mathrm{X})) = \mathbb{E}_\theta(\mathbb{E}_\theta(\mathrm{T}_1(\mathrm{X})|\mathrm{T}(\mathrm{X}))) = \mathbb{E}_\theta \mathrm{T}_1(\mathrm{X})$$

Теорема Рао-Блеквелла-Колмогорова

$$\mathbb{E}_\theta(\mathrm{T}_2(\mathrm{X}) - \theta)^2 \le \mathbb{E}_\theta(\mathrm{T}_1(\mathrm{X}) - \theta)^2$$

$$\mathbb{D}\big(T_1(X)\big) = \mathbb{E}\big(\mathbb{D}(T_1(X)|T(X))\big) + \mathbb{D}\big(\mathbb{E}(T_1(X)|T(X))\big)$$

$$= \mathbb{E}\big(\mathbb{D}(T_1(X)|T(X))\big) + \mathbb{D}\big(T_2(X)\big),$$

$$\mathbb{D}\big(\mathrm{T}_2(\mathrm{X})\big) \le \mathbb{D}\big(\mathrm{T}_1(\mathrm{X})\big)$$

Теорема Рао-Блеквелла-Колмогорова и многочастичный фильтр

Разделение переменных запроса $X'_t \subset X_t$ и $X''_t \subset X_t$

Обновление модели перехода

$$\mathrm{P}(\mathrm{X}_{t+1}|\mathrm{X}_t) = \mathrm{P}(\mathrm{X}'_{t+1}|\mathrm{X}''_{t:t+1}, \mathrm{X}'_t)\mathrm{P}(\mathrm{X}''_{t+1}|\mathrm{X}''_t),$$

Апостериорное распределение для среза T+1

$$\mathrm{P}(\mathrm{X}'_{t+1}, \mathrm{X}''_{t+1}|\mathrm{E}_{1:t+1}) = \mathrm{P}(\mathrm{X}''_{t+1}|\mathrm{E}_{1:t+1})\ \mathrm{P}(\mathrm{X}'_{t+1}|\mathrm{X}''_{t:t+1}, \mathrm{X}'_t)\mathrm{P}(\mathrm{X}''_{t+1}|\mathrm{X}''_t)$$

Весовое распределение для среза T+1

$$\mathrm{W}(\mathrm{X}'_{t+1}, \mathrm{X}''_{t+1}|\mathrm{E}_{1:t+1}) = \mathrm{P}(\mathrm{E}_{t+1}|\mathrm{X}''_{t+1}, \mathrm{X}'_{t+1})\mathrm{N}'(\mathrm{X}''_{t+1}, \mathrm{X}'_{t+1}|\mathrm{E}_{1:t})$$

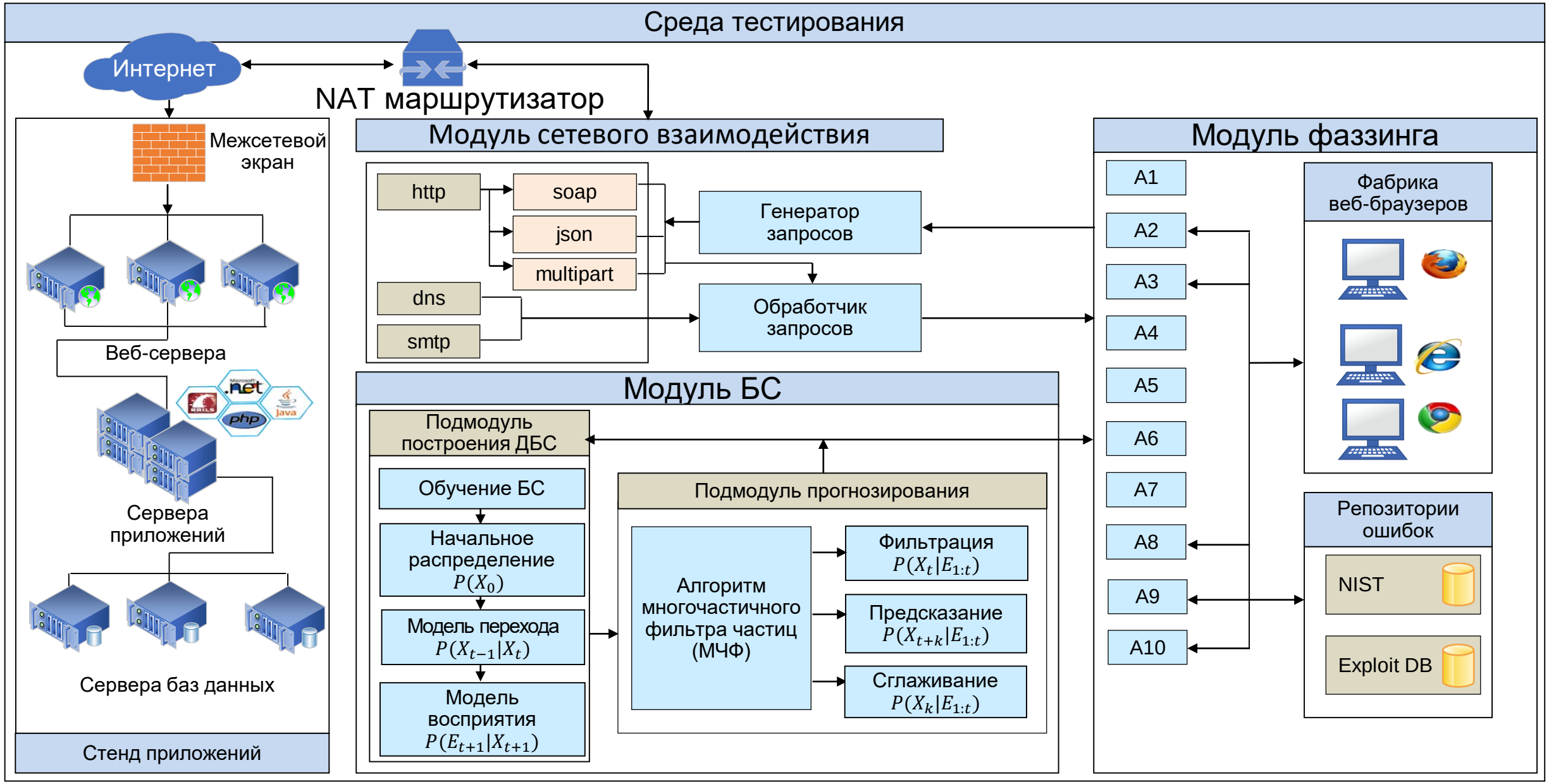
Применение теоремы Рао-Блеквелла-Колмогорова

$$\mathbb{D}(\mathrm{W}(\mathrm{X}'_{t+1}|\mathrm{E}_{1:t+1})) \le \mathbb{D}(\mathrm{W}(\mathrm{X}'_{t+1}, \mathrm{X}''_{t+1}|\mathrm{E}_{1:t+1}))$$

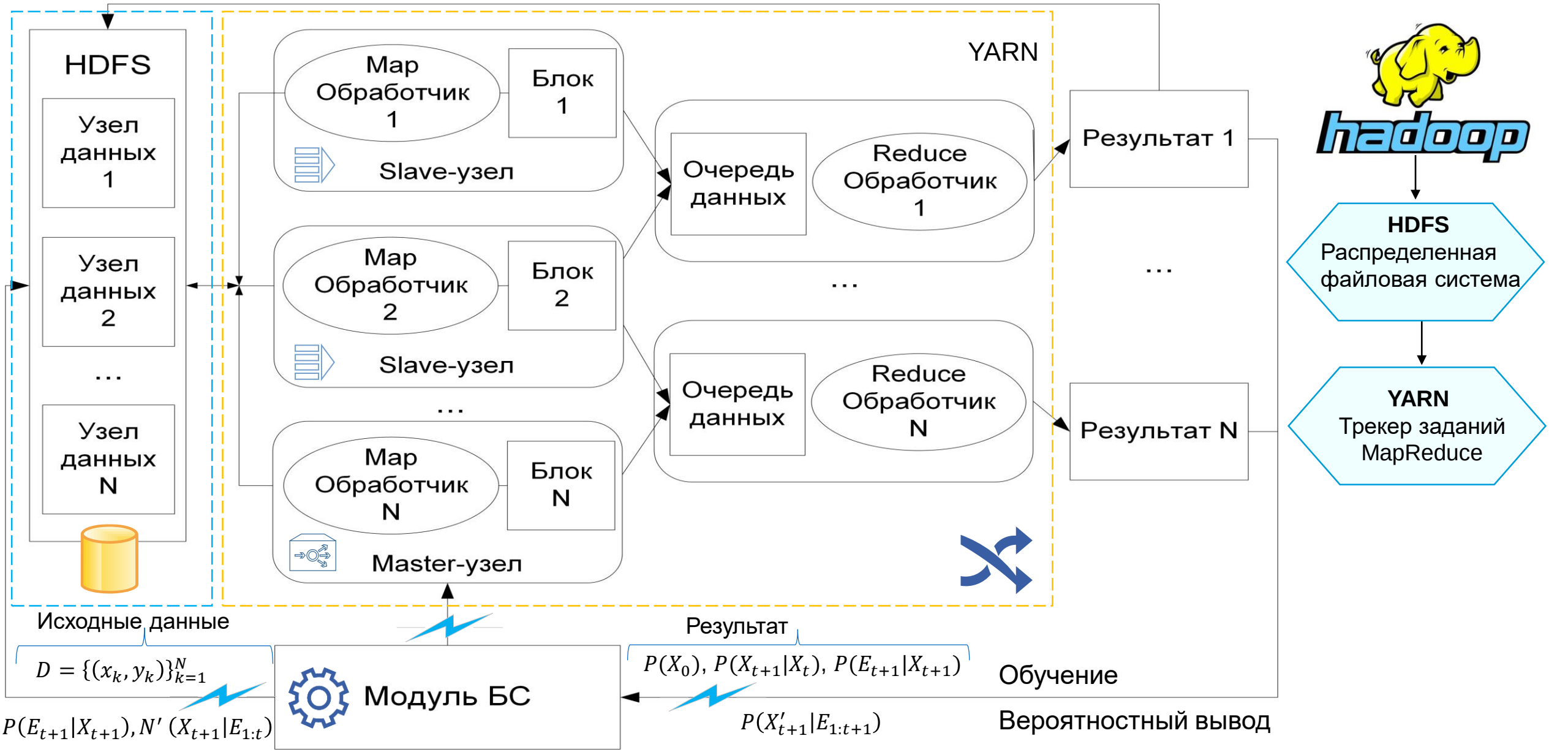
Искомое распределение вероятностей по всем выборкам

$$\mathrm{P}(\mathrm{X}'_{t+1}|\mathrm{E}_{1:t+1}) = \mathrm{N}'(\mathrm{X}''_{t+1}, \mathrm{X}'_{t+1}|\mathrm{E}_{1:t+1})/\mathrm{N}$$

Структура процесса тестирования методом фаззинга



Обучение и вероятностный вывод в ДБС с использованием параллельной платформы Apache Hadoop MapReduce



Временные показатели алгоритмов обучения и вероятностного вывода ДБС «Инъекции»

Аппаратная конфигурация вычислительной системы Apache Hadoop из 6 узлов:

- 6 x (2 процессора Intel Xeon-Platinum 2.5 GHz x 16 ядер, 128 GB ОЗУ);
- 6 x (жесткий диск 10 TB). Размер распределенной файловой системы (HDFS) 59.5 TB;
- оптико-волоконный канал связи между узлами с пропускной способностью до 16Gb/s;

Сравнительные показатели алгоритмов обучения: возрастания-сокращения (ВС), инкрементных ассоциаций Марковского покрытия (ИАМП), минимаксного восхождения (ММВ) и гибридного алгоритма МПМЦ

№ п/п	Размер обучающей выборки D	Алгоритм ВС	Алгоритм ИАМП	Алгоритм ММВ	Алгоритм МПМЦ
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.

Сравнительные показатели алгоритмов вероятностного вывода: Метраполиса-Гастингса (МГ), выборки по значимости (ВЗ), взвешивания с учетом правдоподобия (ВСП), МЧФ и разработанного гибридного алгоритм МЧФ с применением теоремы РБК

№ 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.

Основные результаты

- Построены модели динамических байесовских сетей тестирования методом фаззинга основных классов ошибок веб-приложении.
- Разработаны гибридные алгоритмы обучения, фильтрации, прогнозирования и сглаживания для разработанных динамических байесовских сетей, адаптированные к процедурам тестирования методом фаззинга, реализованные в рамках марковских предположения об условно-вероятностных связях между срезами сети, использующие многочастичный фильтр с применением теоремы Рао-Блеквелла-Колмогорова для оптимизации процедур вероятностного вывода.
- Разработана структура программного обеспечения, содержащего три крупных модуля: модуль сетевого взаимодействия обработки запросов; модуль фаззинга; модуль реализации процесса обучения, вероятностного вывода, фильтрации, предсказания для динамических байесовских сетей тестирования веб-приложений методом фаззинга.

Благодарим за внимание!