ЧЕБЫШЕВСКИЙ СБОРНИК
Том 22. Выпуск 5.

УДК 519.7 DOI 10.22405/2226-8383-2021-22-5-391-399

Скрытые марковские модели в задаче обнаружения атак на компьютерные сети

В. Л. Токарев

Токарев Вячеслав Леонидович — доктор технических наук, профессор, Тульский государственный университет (г. Тула).
email: tokarev22@yandex.ru

Аннотация

В статье рассматривается задача обнаружения атак на компьютерные сети. Предлагается метод проактивного противодействия, основанный на использовании детекторов, выстраиваемых в виде скрытых марковских моделей.

Ключевые слова: Математические модели, информационная безопасность.

Библиография: 7 названий.

Для цитирования:
В. Л. Токарев. Скрытые марковские модели в задаче обнаружения атак на компьютерные сети // Чебышевский сборник. 2021. Т. 22, вып. 5, С. 391–399.

CHEBYSHEVSKII SBORNIK
Vol. 22. No. 5.

UDC 519.7 DOI 10.22405/2226-8383-2021-22-5-391-399

Hidden markov models in the problem of detecting attacks on computer networks

V. L. Tokarev

Tokarev Vyacheslav Leonidovich — doctor of technical sciences, professor, Tula State University (Tula).
email: tokarev22@yandex.ru

Abstract

The article deals with the problem of detecting attacks on computer networks. A method of proactive counteraction based on the use of detectors built in the form of hidden Markov models is proposed.

Keywords: Mathematical models, information security.

Bibliography: 7 titles.

For citation:
V. L. Tokarev, 2021, “Hidden markov models in the problem of detecting attacks on computer networks” , *Chebyshevskii sbornik*, vol. 22, no. 5, pp. 391–399.

1. Введение

На сегодняшний день актуальной задачей является поиск наиболее эффективных методов выявления аномалий в работе компьютерной сети, являющихся следствием технических сбоев, изменений условий работы сети или несанкционированных воздействий. Особую опасность для целостности и конфиденциальности информации, обрабатываемой в сети представляют именно несанкционированные воздействия [1] – атаки на компьютерной сети (КС).

Системы обнаружения атак обычно являются первой линией обороны защищаемой компьютерной системы. Их эффективность существенно зависит от методов обнаружения, на которых они основаны. Большинство таких методов можно разделить на следующие три класса [2].

1) Сигнатурные методы широко используется для обнаружения атак известных видов. Хотя он очень эффективны, но становятся неэффективными, если есть даже очень небольшое изменение в аномалии изменяет сигнатуру атаки. Кроме того, этот метод требует обновления базы данных сигнатур на регулярной основе, чтобы обнаруживать новые разновидности атак [3].

2) Поведенческие методы обеспечивают постоянное наблюдение за поведением КС, чтобы определить, является ли поведение аномалией. Эти методы могут обнаруживать неизвестные атаки. Однако практика показывает, что эти методы имеют высокий уровень ложных срабатываний [4]. Кроме этого, эти методы требуют большого времени процесса обнаружения.

3) Эвристические методы, которые для выявления аномалий используют, в основном, машинное обучение и интеллектуальный анализ данных. Эффективность этих методов напрямую зависит от используемых детекторов – обучаемых программных средств, предназначенных для выявления аномалий.

Очевидно, что в связи с неуклонным расширением множества методов, реализации атак, наблюдаемым в настоящее время, необходимо расширять разнообразие методов обнаружения атак. Актуальность новых подходов к реализации эвристических методов связана еще и с тем,что сложность задачи обнаружения новых видов атак требует применения и более сложного математического аппарата, в частности алгоритмов обучения систем обнаружения атак.

2. Обучаемые детекторы

Предлагаемый метод обнаружения и классификации атак основан на создании и обучении детекторов. Подобные детекторы, основанные на искусственных нейронных сетях, рассмотрены в [2]. В предлагаемом решении использованы скрытые марковские модели (СММ) и механизмы построения детекторов: генерация и обучение детекторов, отбор детекторов и формировние памяти системы обнаружения атак на их основе [5].

Построение такой системы гипотетически можно представить следующим образом.

Обучаемые детекторы генерируются как некоторые модели со случайными начальными значениями параметров, что дает возможность создания большого количества разнообразных по своей структуре детекторов. К основным параметрам отнесены: время жизни детектора, на протяжении которого он может существовать и уровень доверия.

Далее детекторы проходят стадию обучения, на которой они приобретают способность корректно реагировать на появление признаков аномалий в работе защищаемой компьютерной системы (КС).

Затем они отбираются: те из них, которые не обучились правильно классифицировать аномалии, удаляются, а на его место приходит новый, отличный по начальным параметрам, детектор. Иначе, если детектор обнаружил аномалию и правильно классифицировал ее как

атаку, происходит информирование администратора КС об обнаруженной атаке и реакция на нее.

Детектор, обнаруживший атаку, трансформируется в обученный детектор и его первоначально заданные параметры изменяются: 1) увеличивается время жизни и 2) повышается уровень доверия. Отобранные детекторы допускаются к выполнению функций классификации аномалий в поведении КС.

3. Скрытые Марковские модели

В рассматриваемом решениии в качестве обучаемого детектора выбрана СММ. Скрытую марковскую модель можно рассматривать как статистическую модель [6], способную имитировать работу процесса, похожего на марковский процесс с неизвестными параметрами (рис.1).

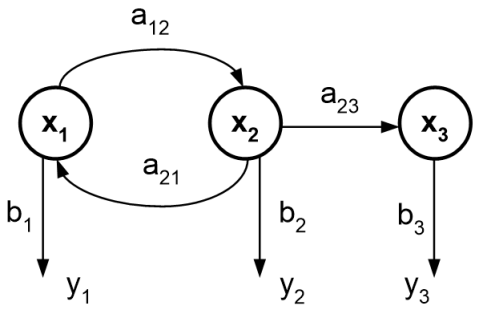

Рис. 1: Пример диаграммы переходов в скрытой Марковской модели: x — скрытые состояния, y — наблюдаемые результаты, a — вероятности переходов, b — вероятность результата

То есть, это вероятностная модель множества случайных переменных $\{y_1, \dots y_t, x_1, \dots x_t, \}$, среди которых y_t — известные дискретные наблюдения, а x_t — «скрытые» дискретные величины. Тогда задачей становится определение неизвестных параметров СММ на основе наблюдений.

Для динамического процесса СММ можно представить в виде связного графа последовательностей (рис. 2).

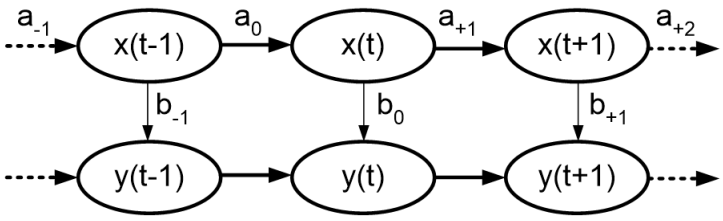

Рис. 2: Связный граф последовательностей

Вероятность наблюдать последовательность длины L равна $P(Y) = \sum P(Y|X) \cdot P(X)$, причем сумма пробегает по всем возможным последовательностям скрытых узлов

Существуют, по крайней мере, два алгоритма, позволяющих решить задачу:

Алгоритм прямого-обратного хода: даны параметры модели B и после-довательность Y, требуется вычислить вероятность A появления данной последовательности X (помогает решить задачу).

Алгоритм Баума-Велша [7]: дана выходная последовательность Y (или не-сколько) с дискретными значениями, требуется «потренировать» СММ на данном выходе (решает задачу).

Алгоритм «прямого-обратного» хода — алгоритм для вычисления апостериорных вероятностей $a \in A$ последовательности состояний $x \in X$ при наличии последовательности наблюдений Y. Иначе говоря, алгоритм, вычисляющий вероятность специфической последовательности наблюдений Y.

Алгоритм включает три шага: 1) вычисление прямых вероятностей; 2) вычисление обратных вероятностей; 3) вычисление сглаженных значений.

Прямые и обратные шаги являются «прямым проходом по сообщению» и «обратным проходом по сообщению», где сообщением выступает ряд последовательных наблюдений Y. Различие в способе, которым алгоритм обрабатывает данную последовательность наблюдений Y.

При прямом обходе, алгоритм продвигается, начинаясь с первого наблюдения в последовательности до последнего, и затем возвращается назад к первому. При каждом наблюдении, вычисляются вероятности, которые будут использоваться для вычислений при следующем наблюдении.

Во время обратного прохода алгоритм одновременно выполняет шаг сглаживания — это процесс вычисления распределения вероятностей значений переменных от прошлых состояниях при наличии свидетельств вплоть до нынешнего состояния. Этот шаг позволяет алгоритму принимать во внимание все прошлые наблюдения для того, чтобы вычислить более точные результаты.

Для представления алгоритм «прямого-обратного» хода удобно использовать матрицу переходных вероятностей $P(X_t|X_t-1)$, представляющей все возможные состояния в СММ данной случайной переменной X_t представить в виде матрицы T, в которой индекс строки i обозначает начальное состояние, а индекс столбца j - конечное состояние.

Тогда следующую прямую вероятность можно вычислить таким образом. Набор прямых вероятностей будем сохранять в ещё одной матрице , где указывает на то, что вычисленные вероятности зависят от всех прямых вероятностей от до , включая текущую матричную вероятность, которую мы опишем как $F_1 : t$.

Следовательно, $F_{1;t+1}$ равно произведению транспонированной матрицы с текущими прямыми вероятностями и матрицей наблюдения для следующего свидетельства в потоке наблюдения. После этого получается матрица, которая требует нормализации, то есть полученные значения должны быть разделены на сумму всех значений в матрице, что можно отобразить коэффициентом нормализации a. Тогда вычисление прямых вероятностей можно описать формулой:

$$F_{1;t+1} = a \cdot Y_{t+1} \cdot T \cdot F_{1;t}$$

Вычисление обратной вероятности $B_{k+1:t}$, можно производить аналогичным способом. Пусть конец последовательности будет описан индексом , начинающийся с 0. Поэтому выполнение от k и вычисляя каждую обратную вероятность можно описать формулой:

$$B_{1:t+1} = T \cdot Y_{k+1} \cdot B_{k+2:t}$$

При этом используется не обычное матричное произведение, а поточечное: умножается значение каждой переменной в одной матрице с соответствующей переменной в другой.

Третий и конечный шаг — это вычисление сглаженных вероятностей . Сглаженные вероятности получаются также поточечным произведением по формуле:

$$C_k = a \cdot B_{k+1:t} \cdot F_{1:k}$$

Для растущего t существуют алгоритмы эффективного вычисления с помощью онлайн сглаживания с фиксированым лагом.

Алгоритм Баума — Велша, предлагаемый для нахождения неизвестных параметров X скрытой марковской модели, использует алгоритм прямого - обратного хода. Его сходимость опирается на два независимых утверждения:

1) скрытая переменная при известной i-ой переменной независима от всех предыдущих (переменных, то есть $P(x_t|x_{t-1}, x_{t-2}, ..., x_1, y_1) = P(x_t|x_{t-1})$;

2) известное наблюдение зависит только от t-го состояния, то есть не зависит от времени: $P(y_t|x_{t-1}, x_{t-2}, ..., x_1, y_1) = P(y_t|x_t)$.

Можно задать $P(q_t|q_{t-1}$ как независящую от времени стохастическую матрицу перемещений $A = \{a_{ij}\} = P(Q_t = j, Q_{t-1} = i)$, где q — дискретная случайная переменная, принимающая одно из значений $(1, \ldots, N)$. Для времени $t = 1$ может быть определено начальное распределение $\pi_i = P(q_1 = i)$.

Будем считать, что процесс находится в состоянии q_t в момент времени t, если $q_t = j$. Последовательность заданных состояний определяется как $q = q_1, ..., q_T$, где $q_t \in 1, ..., N$ является состоянием в момент времени t.

Наблюдение может иметь одно из возможных значений $Y_t = \{y_1, ..., y_L\}$. Вероятность заданного вектора наблюдений в момент времени t для состояния q_t определяется как $b_j(y_t) = P(Y_t - y_t|q_t = j)$, $B = \{b_{ij}\}$ — матрица. Заданная последовательность наблюдений выражается как $Y = Y_1 = y_1, ..., Y_T = y_T$. Тогда СММ можно описать

$$\lambda = (A, B, \pi)$$

При заданном векторе наблюдений алгоритм Баума — Велша находит $\lambda^* = maxP(Y|\lambda)$.

Предлагаемый алгоритм отличается многократной повторной оценкой параметров A, B, π [8]. Число состояний N и число уникальных наблюдаемых символов M являются постоянными. Изменяются на каждом шаге A,B и π.

Алгоритм итеративно обновляет их значения до схождения в одной точке и заключается в следующем:

Инициализировать $s = (A, B, \pi)$, задавая случайные величины, например: $\pi = 1/N, a_{ij} = 1/N, b_{ij} = 1/M$. Прямая процедура:вычисляются рекурсивно.

$$a_i\{1\} = \pi_i \cdot b_i(Y_1)$$

$$a_j(t+1) = b_j(Y_{t+1}) \cdot \sum_{i=1}^{N} a_i(t) \cdot a_{ij}$$

Обратная процедура:

$$beta_i(T) = P(Q_t = i, \lambda) = 1$$

$$beta_i(t) = \sum_{i=1}^{N} beta_i(t+1) \cdot a_{ij} \cdot b_j(Y_{t+1})$$

$$\pi_i = \frac{\alpha_i(1) \cdot \beta_i(1)}{\sum_{j=1}^{N} \alpha_j(1) \cdot \beta_j(1)}$$

Используя новые значения векторов A, B и π, итерации продолжаются до их схождения.

4. Обучение и отбор СММ-детекторов

СММ-детекторы $d \in D$, проходят стадии своего жизненного цикла τ следующим образом. Генерация множества детекторов $D(W_{ij}, \tau) \to a_{ij}$ (каждое множество $D(W_{ij}, \tau)$ соответствуют одному типу атаки a_{ij}. Это создание СММ с начальной инициализацией вектора π_0 по случайному закону.

Обучающая выборка training-set содержит множество параметров защищаемой КС, разделенное на две части – одна training-set-a_{ij}, содержащая параметры нормального поведения КС (при полном отсутствии какой – либо аномалии), другая training-set-n - параметры соединений, соответствующие конкретной атаке. При этом $D_{ij}(W_{ij}) \subset D$, $|D(W_{ij})| = k_d$, $|D| = k_d/cdotk_a$.

Правило обучения заключается в том, чтобы настроить нужным образом вероятности a,b.

Результат такого обучения состоит в том, что победивший детектор с большей вероятностью выиграет конкуренцию и в том случае, когда на вход поступит новый вектор $p(l+1)$, близкий предыдущему $p(l)$, и с меньшей вероятностью, когда будет получен новый вектор $p(l+1)$, существенно отличающийся от $p(l)$. При большом количестве поступающих векторов $p(l), l = 2, 3, \ldots$ детектор v_i корректирует свои параметры (векторы A и B). В этом и заключается самоорганизация СММ- детектора.

При такой коррекции векторов A и B некоторые детекторы остаются незадействованными (детекторы с изначально удаленными параматрами) никогда не выигрывают конкуренции, независимо от того, как долго продолжается обучение. В результате оказывается, что такие векторы w_i не используются при обучении и не выполняют никакой полезной функции. Чтобы исключить такие ситуации и сделать детекторы чувствительными к поступающим векторам $p(l), l = 2, 3, \ldots$ используются смещения, которые позволяют детектору v_i стать конкурентным с детектором w_j.

Использование таких смещений позволяет: 1) вовлечь в процесс обучения больше детекторов, 2) выравнивать значения праметров активности и обеспечить притяжение примерно одинакового количества векторов входа p.

Далее процесс повторяется, начиная с третьего шага для всех входных данных.

Обучение СММ-детекторов производится до желаемой степени согласования между входными и весовыми векторами, т. е. до тех пор, пока значение суммарной квадратичной ошибки не станет равной нулю и весь процесс повторяется до тех пор, пока количество обученных иммунных детекторов не станет равным заданному значению k_ij.

Таким образом, создается набор детекторов для анализа поведения КС. Однако перед тем, как выполнять анализ КС, обученные детекторы необходимо проверить на корректность классификации с целью предотвращения возникновения ложных срабатываний. Для этого все обученные детекторы проходят стадию отбора.

Отбор СММ-детекторов по тестовой выборке. Для минимизации возникновения ложных срабатываний, когда нормальное соединение принимается за компьютерную атаку, все обученные детекторы проходят проверку на корректность классификации. Для этого детектор подается заранее созданная тестовая выборка, состоящая из параметров нормального соединения. Если i-й детектор классифицирует одно из тестовых соединений как атаку, то он уничтожается, а вместо него генерируется и обучается новый детектор. Если i-й детектор не генерирует ложные срабатывания на тестовой выборке, то он считается корректным и допускается к анализу входящего и исходящего сетевого трафика.

5. Функционирование обученных детекторов

Детекторы, которые допущены к анализу поведения КС, образуют систему обнаружения атак.

Механизм наделения детекторов временем жизни позволяет избавляться от детекторов, которые хоть и прошли успешно стадии обучения и отбора, однако из-за своей особенности (векторы A и B) являются малопригодными.

Активация детекторов. Активация детекторов подразумевает обнаружение детектором атаки. В случае, когда запущенная программа классифицируется одним или несколькими детекторами как атака, происходит её блокировка. Также выдается сообщение пользователю об обнаруженю атаки.

Формирование памяти системы обнаружения атак. При обнаружении и блокировании атаки целесообразно сохранять ее параметры с целью изучения и детального анализа. Дело в том, что детекторы обучаются на ограниченном наборе данных, которые не могут включать в себя все вероятные атаки. Для того чтобы повысить качество обнаружения, а также наделить систему гибкостью и позволить ей адаптироваться под современные реалии, параметры аномалии, классифицированной как атака, сохраняются и заносятся в обучающую выборку, тем самым пополняя ее актуальными данными.

Детекторы, которые будут создаваться с целью заменить «устаревшие» детекторы, будут уже обучаться также и на новых данных, что позволит значительно увеличить качество обнаружения. Кроме этого, создается новая СММ, которая обучается исключительно на данных, выделенных из обнаруженной атаки, и которая вводится в систему анализа аномалий. Это позволит более точно выделить данную атаку при повторной подобной атаке на защищаемую коппьютерную систему со стороны злоумышленника.

Совокупность детекторов (память системы обнаружения) хранит в себе информацию обо всех атаках, направленных в прошлом на защищаемую автоматизированную систему, и обеспечивать высокий уровень реагирования на повторные попытки реализации атаки.

Общий *алгоритм функционирования системы обнаружения атак* на базе обучаемых детекторов можно представить в следующем виде:

1. Создание детектора на базе СММ и начальная инициализация вектора параметров π.

2. Затем формируется обучающая выборка, включающая n значений из множества Y, соответствующего нормальному функционированию автоматизированной системы, и n значений из множества Y*, соответствующих атаке.

3. Последовательно подавая данные из обучающей выборки на СММ и обучаем ее согласно правилу «победитель берет все».

4. Обученный детектор проверяется на тестовой выборке, состоящей из параметров A и B СММ, соответствущей нормальному состоянию КС. Если детектор корректно классифицирует предоставленные данные, то он «допускается» к анализу аномалий в реальном режиме. Если же детектор классифицирует предоставляемые тестовые данные как атаку, то он уничтожается.

5. Внедрение обученного и отобранного детектора в подсистему анализа системмы обнаружения атак.

6. Если отведенное время жизни детектора истекло и детектор не обнаружил аномалию в поведении КС, то он уничтожается и процесс начинается с шага 1. Если детектор обнаружил аномалию, то происходит его активация. Аномалия, классифицированная как атака, блокируется, а администратору КС выдается сообщение.

7. Выделение и анализ параметров аномалии в поведении КС, классифицированной как атака и занесение выделенных параметров в базу для обучения новых иммунных детекторов.

Экспериментальные исследования СММ-детекторов в задачах выявления атак показали: 1) их достаточно высокую точность обнаружения (в среднем 91%); 2) процесс обнаружения атак СММ-детекторами занимает меньшее время по сравнению с детекторами на нейронной сети Кохонена в среднем на 6%.

6. Заключение

В настоящем исследовании рассмотрен метод использования СММ-детекторов для обнаружения атак и сделан вывод, системы для обнаружения атак, построенные на основе СММ-детекторов, достаточно эффективны, и дальнейшие исследования могут быть сосредоточены на том, как снизить количество ложных «срабатываний» на аномалии, которые не являются результатом атаки.

СПИСОК ЦИТИРОВАННОЙ ЛИТЕРАТУРЫ

1. Bazrafshan, Z., Hashemi, H., Fard, S.M.H. and Hamzeh, A. (2013) A Survey on Heuristic Malware Detection Techniques. The 5th Conference on Information and Knowledge Technology (IKT 2013), Shiraz, 28-30 May 2013, 113-120. http://dx.doi.org/10.1109/ikt.2013.6620049

2. Шелухин О.И. Обнаружение вторжений в компьютерные сети (сетевые аномалии)/ О.И. Шелухин, Д.Ж. Сакалема, А.С. Филинова – М.: Горячая линия – Телеком, 2013. – 220 с.

3. Бирюков А.А. Информационная безопасность: защита и нападение. – М.: ДМК Пресс, 2012. – 474 с.

4. Крис Касперски. Компьютерные вирусы изнутри и снаружи. – СбП: Питер, 2006. – 526 с.

5. Токарев В.Л., Сычугов А.А. Обнаружение вредоносного программного обеспечения с использованием иммунных детекторов // Известия Тульского государственного университета. Технические науки. Вып. 10. Тула: Изд-во ТулГУ, 2017. - с.216-230.

6. Alqurashi, S. and Batarfi, O. A Comparison of Malware Detection Techniques Based on Hidden Markov Model. Journal of Information Security, 7, 215-223. http://dx.doi.org/10.4236/jis.2016.73017

7. Frazzoli, Emilio. "Intro to Hidden Markov Models the Baum-Welch Algorithm". Aeronautics and Astronautics, Massachusetts Institute of Technology. Retrieved 2 October 2013.

REFERENCES

1. Bazrafshan, Z., Hashemi, H., Fard, S.M.H. and Hamzeh, A. (2013) A Survey on Heuristic Malware Detection Techniques. The 5th Conference on Information and Knowledge Technology (IKT 2013), Shiraz, 28-30 May 2013, 113-120. http://dx.doi.org/10.1109/ikt.2013.6620049

2. Shelukhin O. I. Detection of intrusions into computer networks (network anomalies) / O. I. Shelukhin, D. Zh. Sakalema, A. S. Filinova-M.: Hotline-Telecom, 2013. - 220 p

3. Biryukov A. A. Information security: protection and attack. - Moscow: DMK Press, 2012. - 474 p.

4. Chris Kaspersky. Computer viruses inside and out. - SbP: Peter, 2006. - 526 p.

5. Tokarev V. L., Sychugov A. A. Detection of malicious software using immune detectors / / Izvestiya Tulskogo gosudarstvennogo universiteta. Technical sciences. Issue 10. Tula: TulSU Publishing House, 2017. - pp.216-230.

6. Alqurashi, S. and Batarfi, O. A Comparison of Malware Detection Techniques Based on Hidden Markov Model. Journal of Information Security, 7, 215-223. http://dx.doi.org/10.4236/jis.2016.73017

7. Frazzoli, Emilio. "Intro to Hidden Markov Models the Baum-Welch Algorithm". Aeronautics and Astronautics, Massachusetts Institute of Technology. Retrieved 2 October 2013.

Получено 14.06.2021 г.
Принято в печать 21.12.2021 г.