АЛГОРИТМ ИЗВЛЕЧЕНИЯ ХАРАКТЕРНЫХ ПРИЗНАКОВ ИЗ ДАННЫХ ПОЛЬЗОВАТЕЛЬСКИХ АКТИВНОСТЕЙ

Слипенчук П.В.[1]

В данной работе представлен алгоритм извлечения признаков (feature extraction) из данных пользовательских активностей. Алгоритм с учителем, требует разделения на два класса. Для работы алгоритма требуются данные пользовательской активности в количестве от миллиона сессий, из которых не менее тысячи были мошенническими. Алгоритм позволяет находить подпоследовательности действий более характерные для мошеннических операций, чем для легитимных и наоборот. Результаты этой работы могут выступать как часть экспертной системы в качестве feature extraction модуля и для предварительного анализа информативности данных. Алгоритм был протестирован на реальных данных крупного ДБО (дистанционного банковского обслуживания) одного из заказчиков продукта Secure Bank и показал высокие результаты. В отличие от большинства других способов извлечения признаков, рассматриваемый алгоритм имеет линейную сложность по построению и по требуемой памяти. Параметры алгоритма интуитивно понятны и легко подбираются профессионалом в области машинного обучения на данных заказчика. Несмотря на то, что алгоритм тестировался в первую очередь для детектирования мошеннических действий в ДБО, алгоритм может быть применён в любых иных данных пользовательской активности в качестве метода извлечения признаков для двухклассовой классификации.

Ключевые слова: *информационная безопасность, социальная инженерия, банковский фрод, UEBA, ДБО, извлечение признаков, feature extraction, пользовательская активность.*

DOI: *10.21681/2311-3456-2019-1-53-58*

Введение

Поведенческая активность на web сайтах всё чаще используется для дополнительной аутентификации пользователей [16, 17]. В настоящий момент данные системы не очень надёжны, однако могут выступать в качестве дополнительного метода аутентификации пользователей. Например, вместо двухфакторной аутентификации [9] на Web-порталах и Web-приложениях, представляющих собой введение пароля и подтверждение по SMS, можно предложить двухфакторную аутентификацию, основанную на введении пароля и анализа поведения пользователя в системе.

Задача анализа этой активности является актуальной проблемой информационной безопасности. Помимо задач информационной безопасности, поведенческая активность может быть использована в рекламе [18].

Существуют решения, основанные на нейронных сетях [10], которые позволяют классифицировать пользователей. Однако при высоких нагрузках, которые существуют для многих Web-приложениях и Web-порталах, они не применимы на практике в режиме online, либо, в лучшем случае, требуют большого количества вычислительных ресурсов.

Поэтому возникает потребность в разработки “легковесных” алгоритмов, позволяющие достаточно эффективно анализировать пользовательские данные. Один из таких алгоритмов приведён в данной статье.

Постановка задачи.

В современных средствах коммуникации (например Web-портал) пользователем совершаются определённые действия в определенных последовательностях. Схематично действия можно представить в виде цепочки действий (см. рис. 1). Это представляет собой однонаправленный граф, из каждой вершины которой исходит не более одного действия. Есть вершина, не имеющая предков — это первое действие пользователя в системе; и вершина, не имеющая потомков — это последнее действие пользователя в системе.

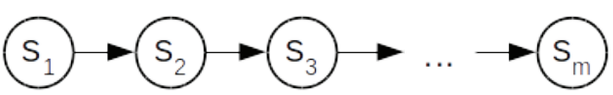

Рис.1. Схематическое изображение действий пользователя в виде цепочки действий

Данный граф можно представить в виде одномерного вектора (s_1, s_2, ..., s_m). Обозначим этот вектор действий за **a**.

$$a = (s_1, s_2, ..., s_m) \quad (1)$$

Заметим, что m не фиксировано. Некоторые пользователи могут совершить в рамках одной сессии большое количество действий (или меньшее).

Предположим, что мы имеем множество таких **a**, от разных пользователей системы. При этом некоторые из этих действий являются легитимными, а некоторые мошенническими.

Обозначим легитимные действия флагом **f=0**, а мошеннические флагом **f=1**. Таким образом имеем множество пар вида:

$$(a_1, f_1), (a_2, f_2), ..., (a_n, f_n) \quad (2)$$

Определим A как множество всевозможных a, для которых известна пара (2). Необходимо создать экспертную систему **F(A)**, которая выдаст два множества: наиболее характерных подпоследовательностей для

1 Слипенчук Павел Владимирович, архитектор систем машинного обучения, ООО «Группа АйБи», Москва, Россия. E-mail: slipenchuk@group-ib.ru

f=0 и для **f=1**. Обозначим эти подпоследовательности за b.

$F(A) = (\{b_1, b_2, ..., b_x\}, \{b_{x+1}, b_{x+2}, ... , b_{y-x}\})$ (3)

где x, и y — параметры системы, задаваемые разработчиком вручную. Параметр **x** — это количество характерных подпоследовательностей для **f=0**, а параметр y — это количество характерных подпоследовательностей для **f=1**.

Формальное описание алгоритма.

Задачу можно пытаться решить различными классическими способами. Экспертные системы, основанные на цепях Маркова [6], не дали хороших практических результатов.

Логично предположить, что данная задача может быть решена свёрточными нейронными сетями [3], однако в контексте решаемой нами задачи мы даже не рассматривали подобный вариант. Дело в том, что у продукта Secure Bank огромная нагрузка (более 15000 POST запросов в секунду, более 3 ГБ трафика в секунду) и требовалось “в режиме online” выносить вердикт [3]. Однако для задач, где требуется вынесение решений постфактум, идея со свёрточными сетями разумна.

Было принято решение на создание собственного алгоритма, выдающего наиболее характерные подпоследовательности.

Зафиксируем целочисленные l_1 и l_2.

Это параметры системы, задаваемые разработчиком системы вручную. Из любого вектора a можно получить множество различных векторов длины не более l_1 и не менее l_2. Возможные последовательности получаемые из ai обозначим за bi,j. Функцию, выдающее множество {bi,j} обозначим за B.

$B(a_i, l_1, l_2) = \{b_{i,j}\}$ (4)

Определим величину **C(b,f)** как количество a из **A**, которые содержат подвектор **b** и которым соответствует флаг **f** (см (2)).

Определим q как количество **a** из **A** с флагом **f=0**, делённое на количество **a** из **A** с флагом **f=1**.

Для каждого b зададим функцию веса:

$$W(\mathbf{b}) = \frac{q \cdot C(\mathbf{b};1) + h_1 + h_2 \cdot C(\mathbf{b};0)}{h_3 \cdot C(\mathbf{b};0) + h_1 + 1} \quad (5)$$

В функции (5) веса величины h_1, h_2, h_3 являются параметрами системы и задаются вручную разработчиком.

Если h_1, h_2, h_3 неотрицательны, то можно заметить, что чем **W(b)** больше, тем **b** скорее характеризует мошенническую подпоследовательность действий, чем легитимную. Осталось только найти всевозможные **b**, упорядочить их по весу и выбрать x с наименьшим весом и **y** с наибольшим.

Одним из способов найти все b с заданным весом следующий.

Определим функцию **G(b)** как функцию, разбивающая вектор на множество смежных векторов длины 2:

$G(b) = G(s_i, s_{i+1}, ..., s_{i+j}) = \{ (0, s_i), (s_i, s_{i+1}), ..., (s_{i+j-1}, s_{i+j})\}$ (6)

Построим граф D по следующему алгоритму:

1. Определим D как граф содержащий одну вершину: 0
2. Выбираем любой **a** из **A**
3. Определяем флаг **f** для **a** из (2)
4. Помечаем вершину 0
5. Для каждого **b** из $\mathbf{B(a, l_1, l_2)}$ и для каждой пары $\mathbf{(s_1, s_2)}$ из **G(b)**:

a. Если от вершины помеченной s_1 нет вершины s_2, то создаем вершину s_2 и для ребра (s_1, s_2) устанавливаем метку из двух целочисленных значений: (0, 0).

b. Берем метку ребра (s_1, s_2). Обозначим ее за (c_1, c_2). Переопределяем эту метку:

i. (c_{1+1}, c_2), если f=0

ii. (c_1, c_{2+1}), если f=1

c. Помечаем вершину s_2

6. Удаляем **a** из **A**
7. Если A не пустое множество, иначе п.2
8. Конец алгоритма

В отличие от “классических” графов, у которых каждая вершина уникальна, в нашем случае уникальна только вершина 0. В остальных случаях вершина с одним и тем же именем может встретиться более одного раза. Однако для любой вершины дерева D существует не более одной вершины с определённым именем s_i, поэтому алгоритм построения корректен. Для построения множества всевозможных подпоследовательностей **{b}** нужно выбрать любую вершину и двигаться к вершине 0, записывая имена вершин справа налево. Для определения веса нужно взять ровно одно ребро от вершины, ведущее к вершине 0 (оно всегда одно) и узнать его пару весов (c_1, c_2). В силу построения алгоритма, очевидно, что:

$C(b, 0) = c_1$

$C(b, 1) = c_2$ (7)

Подставляя (7) в формулу (5) получаем вес последовательности b.

Упорядочим все **b** по весу. Выбраем **x** с наименьшим весом и **y** с наибольшим. Формируем пару множеств по (3).

F(A) построена.

Алгоритм имеет линейную сложность построения, что видно из его описания. Поговорим о требуемой памяти. Можно видеть, что оценка сверху имеет экспоненциальный характер и определяется как мощность алфавита {si} в степени $\mathbf{l_2}$:

$O(||\{s_i\}||^{l2})$ (8)

Однако на практике формула (8) не верна. В действительности на практике при больших $\mathbf{l_2}$ получаем линейную зависимость:

$O(l_2)$ (9)

Зависимость сложности для памяти приведена на рисунке 2.

Апробирование результатов.

В качестве примера полезности данной системы, можно привести продукт Secure Bank [2], в котором реализован UEBA [1, 2] модуль, позволяющий обнаруживать пользователей, находящихся под воздействием социальной инженерии [5].

Полученные с помощью описываемого алгоритма признаки можно использовать для алгоритма машинного обучения. Для апробирования результатов был вы-

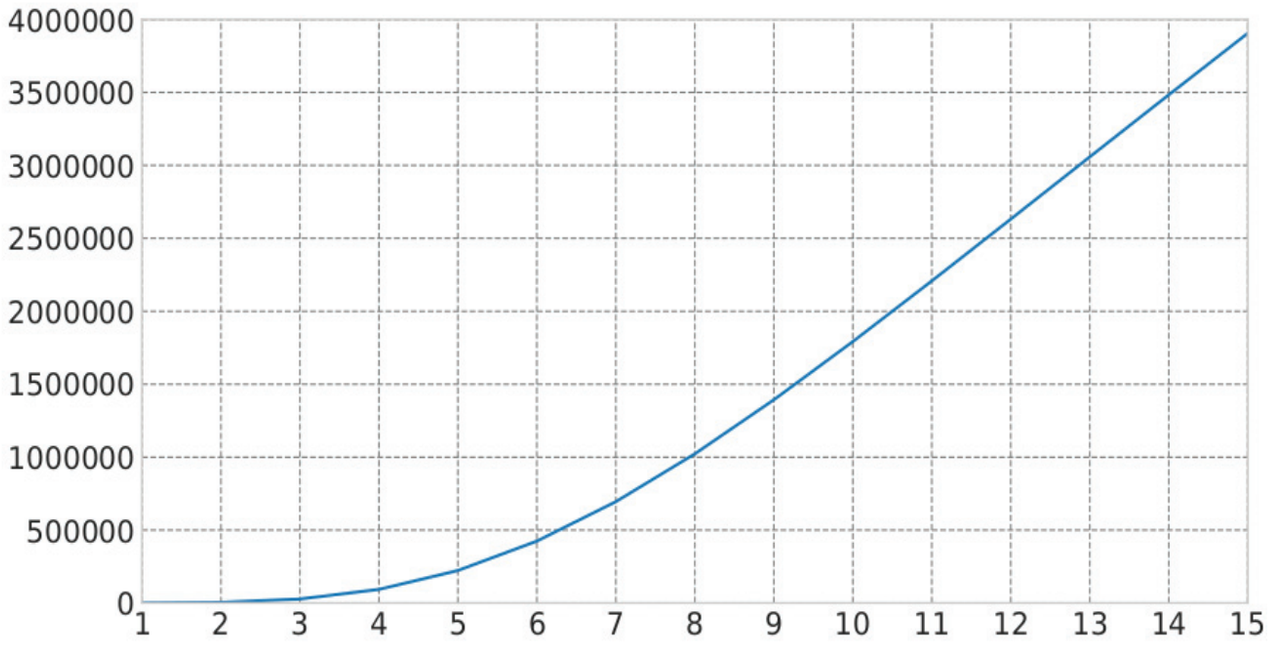

Рис.2. Зависимость количества вершин графа, в зависимости от l2

бран классический алгоритм Random Forest[7, 8, 13]. Данный алгоритм, несмотря на простоту, до сих пор является одним из самых эффективных методов для противодействия мошенничеству в удалённых каналах обслуживания [14, 15].

Для web-портала дистанционного банковского обслуживания одного из клиентов Secure Bank был установлен скрипт, который собирал данные пользовательской активности:

1. Переходы между страницами;
2. Нажатие определённых кнопок интерфейса;
3. Установка и потеря фокуса с объектов (edit, кнопка, картинка и т.д.);
4. Открытие, закрытие вкладки;
5. Факт скроллинга страницы вверх или вниз;

Полнота и точность определялись стандартным способом [11].

В качестве данных были взяты сессии одного из клиентов Secure Bank. Десять миллионов сессий были использованы для извлечения признаков и сто тысяч

Таблица 1

Отсечка	Полнота	Точность
0.1	0.53	0.004
0.2	0.45	0.009
0.3	0.39	0.013
0.4	0.31	0.017
0.5	0.28	0.023
0.6	0.27	0.038
0.7	0.25	0.054
0.8	0.18	0.081
0.9	0.12	0.093

Информативность [12] данных видна на графике для алгоритма машинного обучения. В таблице 1 представлены результаты полноты и точности для одного из крупных заказчиков системы Secure Bank.

Как видно из графика, многие признаки весьма репрезентативны. Каждому признаку присвоен номер, упорядоченный по информативности. “Хвост” с большим количеством нулевых значений показывает, что параметры x и y были выбраны с большим запасом.

На рисунке 4 приведена попарная корреляция признаков.

Это значит, что для отбора признаков разумно удалять признаки с корреляцией равные 1 и близкие к 1. Однако таких признаков не очень много (см. рис.3), не превышает 5%. Следовательно алгоритм извлечения признаков можно считать качественным.

Алгоритм был реализован для крупного заказчика продукта Secure Bank более шести месяцев назад и хорошо зарекомендовал себя. На практике более 70% обнаруживаемого мошенничества было мошенничеством социальной инженерии.

Выводы

1. Приведенный алгоритм извлечения признаков является конструктивным и полезным для решения задачи классификации мошеннических и легитимных транзакций.
2. Даже шаблонный алгоритм машинного обучения (случайный лес) даёт хороший результат по полноте и точности.
3. Коррелируемых признаков (с коэфициентом 0.2 и более) не превышает 5% от всех признаков.
4. Алгоритм имеет линейную сложность по построению и линейную сложность по требуемой памяти.
5. Алгоритм эффективен для детектирования социальной инженерии

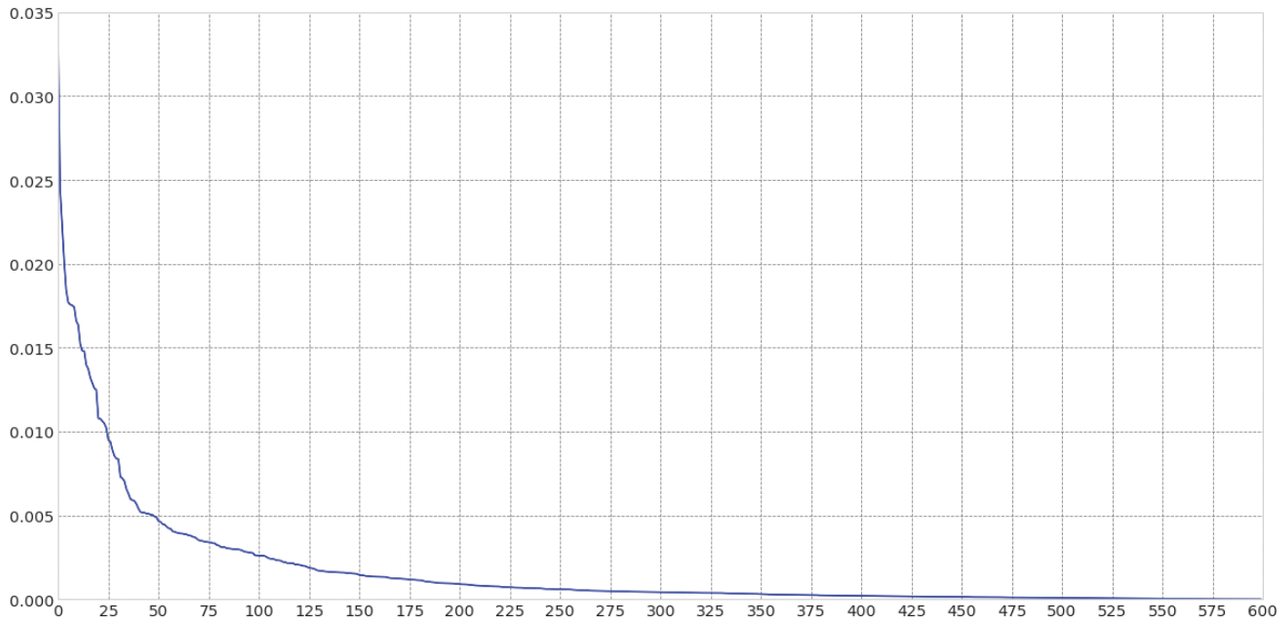

Рис.3. График информативности признаков случайного леса

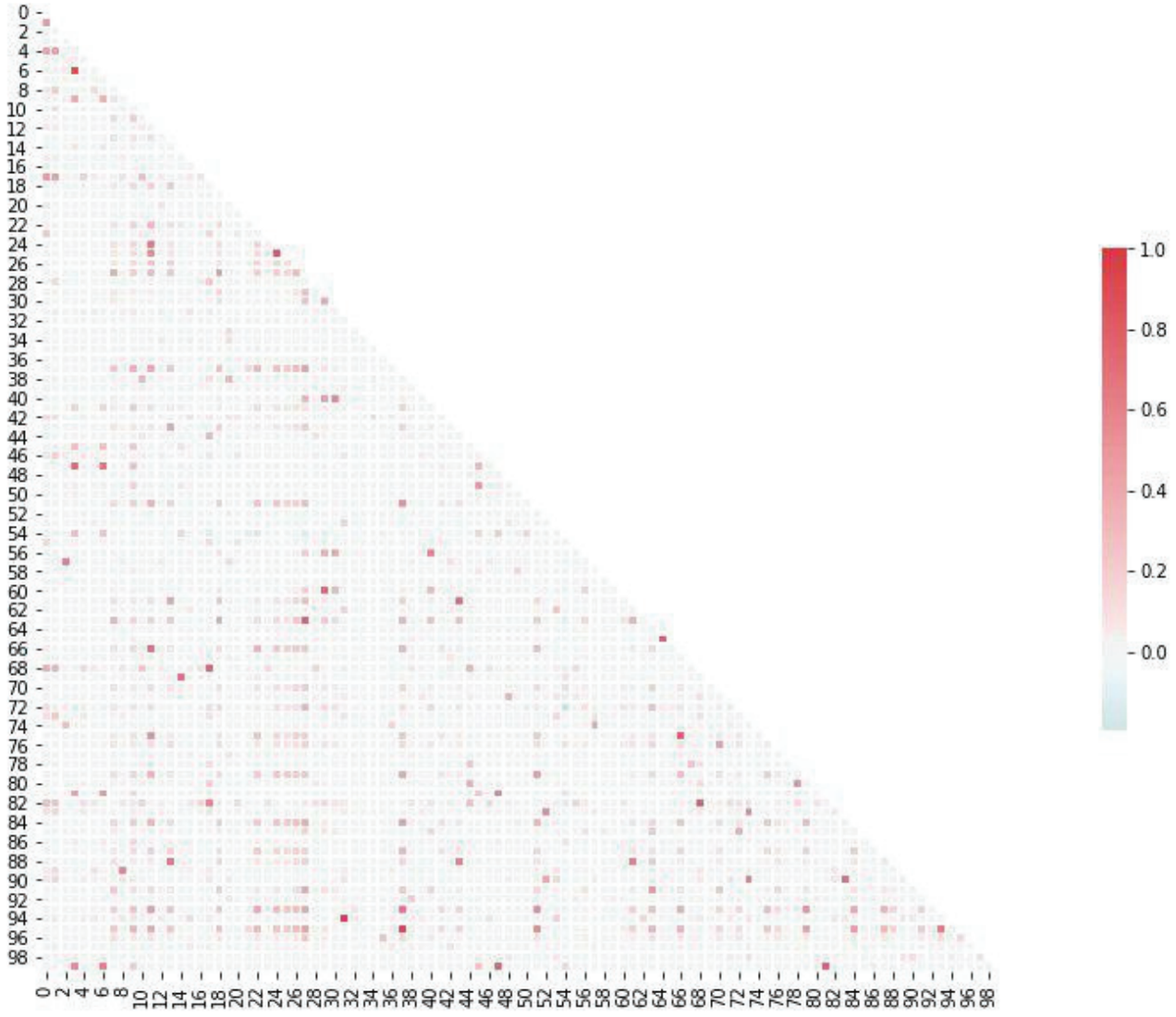

Рис.4. Попарная корреляция первых 100 наиболее характерных признаков

Литература
1. Madhu Shashanka, Min-Yi Shen, Jisheng Wang, «User and entity behavior analytics for enterprise security», 2016. IEEE International Conference on Big Data.
2. Gorka Sadowski, Avivah Litan, Toby Bussa, Tricia Phillips. Market Guide for User and Entity Behavior Analytics Published: 23 April 2018. ID: G00349450, Gartner, 2018.
3. Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, IEEE Transactions on Pattern Analysis and Machine Intelligence (Volume: 39, Issue: 6 , June 1 2017) DOI: 10.1109/TPAMI.2016.2577031.
4. Программный комплекс “Secure Bank”, https://www.group-ib.ru/secure-bank.html (дата обращения 2018.08.20).
5. Radha Gulati «The Threat of Social Engineering and Your Defense Against It», SANS Institute InfoSec Reading Room, 2003.
6. Nong Ye, «A Markov Chain Model of Temporal Behavior for Anomaly Detection», IEEE Workshop on Information Assurance and Security, United States Military Academ, West Point, NY, 6-7 June, 2000.
7. Чистяков С. П. «Случайные леса: обзор», Труды Карельского научного центра РАН, № 1. 2013. С. 117–136.
8. Библиотека sklearn ensemble.RandomForestClassifier, URL: http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html (дата обращения 2018.08.20)
9. Dipankar Dasgupta, Arunava Roy, Abhijit Nag, «Multi-Factor Authentication», Springer «Advances in User Authentication» pp 185-233.
10. Chandrika Palagiri, Rasheda Smith, Alan Bivens. «Network-Based Instrucion detection using neural networks» https://www.researchgate.net/publication/2548475_Network-Based_Intrusion_Detection_Using_Neural (дата обращения 2018.08.20)
11. Tom Fawcett, «An Introduction to ROC Analysis». Pattern Recognition Letters. 27 (8): 861–874. doi:10.1016/j.patrec.2005.10.010, 2006.
12. Miron B. Kursa, Witold R. Rudnicki. Feature Selection with the Boruta Package. Journal of Statistical Software, 36(11), p. 1–13, 2010.
13. S.I. Dimitriadis, Dimitris Liparas, Magda N. Tsolaki, «Random forest feature selection, fusion and ensemble strategy: Combining multiple morphological MRI measures to discriminate among healhy elderly, MCI, cMCI and alzheimer’s disease patients: From the alzheimer’s disease neuroimaging initiative (ADNI) database» Journal of Neuroscience Methods, Volume 302, 2018, Pages 14-23.
14. S. Xuan, G. Liu, Z. Li, L. Zheng, S. Wang and C. Jiang, «Random forest for credit card fraud detection,» 2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC), Zhuhai, 2018, pp. 1-6.
15. Sanaz Nami, Mehdi Shajari, Cost-sensitive payment card fraud detection based on dynamic random forest and k-nearest neighbors, Expert Systems with Applications, Volume 110, 2018, pp. 381-392.
16. Eshghi A, Kargari M. Detecting frauds using customer behavior trend analysis and known scenarios. IJIEPR. 2018, pp. 91-101.
17. Deshpande D., Deshpande S., Thakare V. Analysis of Online Suspicious Behavior Patterns. Ambient Communications and Computer Systems. Advances in Intelligent Systems and Computing, vol 696. Springer, Singapore, 2018, pp. 485-495
18. Gábor Szirtes, Javier Orozco, István Petrás, Dániel Szolgay, Ákos Utasi, Jeffrey F. Cohn, Behavioral cues help predict impact of advertising on future sales, Image and Vision Computing, Volume 65, 2017, pp. 49-57

Рецензент: *Цирлов Валентин Леонидович, кандидат технических наук, доцент кафедры «Информационная безопасность» МГТУ им. Н.Э. Баумана, Москва, Россия.* E-mail: v.tsirlov@npo-echelon.ru

ALGORITHM FOR FEATURE EXTRACTION FROM USER ACTIVITIES DATA

Slipenchuk P.[2]

This paper presents *an algorithm for feature extraction from user activities data. The algorithm with a teacher requires division into two classes. The algorithm requires user activities data in the amount of a million sessions, of which at least a thousand were fraudulent. The algorithm helps find subsequences of actions more typical of fraudulent operations than legitimate ones and vice versa. The results of this work can serve as a feature extraction module (part of the expert system) and for preliminary analysis of data informativeness. The algorithm was tested on real data of a large remote banking service for one of the Secure Bank product customers and showed excellent results. As distinct from most other feature extraction methods, the algorithm under review has linear complexity by construction and by the required memory. The algorithm parameters are intuitive and easily selected by a machine learning professional using customer data. Although the algorithm was tested primarily for detecting fraudulent activities in remote banking services, it can be applied to any other user activities data as a feature extraction method for two-class classification.*

Keywords: *information security, social engeneering, bank fraud, UEBA, online banking, feature extraction, user activity*

References:
1. Madhu Shashanka, Min-Yi Shen, Jisheng Wang. User and entity behavior analytics for enterprise security. In: 2016 IEEE International Conference on Big Data. DOI: 10.1109/BigData.2016.7840805.

2 Pavel Slipenchuk, ML & DM Architect, Group-IB, Moscow, Russia. E-mail: slipenchuk@group-ib.com

2. Gorka Sadowski, Avivah Litan, Toby Bussa, Tricia Phillips. Market Guide for User and Entity Behavior Analytics Published: 23 April 2018, ID: G00349450, Gartner, 2018.
3. Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, IEEE Transactions on Pattern Analysis and Machine Intelligence (Volume: 39, Issue: 6 June 1 2017). DOI: 10.1109/TPAMI.2016.2577031
4. Software platform “Secure Bank”, URL: https://www.group-ib.com/secure-bank.html
5. Radha Gulati. The Threat of Social Engineering and Your Defense Against It, SANS Institute InfoSec Reading Room, 2003.
6. Nong Ye. A Markov Chain Model of Temporal Behavior for Anomaly Detection, IEEE Workshop on Information Assurance and Security, United States Military Academ, West Point, NY, 6-7 June, 2000.
7. Chistiakov C. P. «Sluchai`ny`e lesa: obzor», Trudy` Karel`skogo nauchnogo centra RAN, № 1. 2013. S. 117–136. (In Rus)
8. Sklearn fraimwork: ensemble.RandomForestClassifier, URL: http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html .
9. Dipankar Dasgupta, Arunava Roy, Abhijit Nag. Multi-Factor Authentication, Springer. Advances in User Authentication, pp 185-233.
10. Chandrika Palagiri, Rasheda Smith, Alan Bivens. «Network-Based Instrucion detection using neural networks», URL: https://www.researchgate.net/publication/2548475_Network-Based_Intrusion_Detection_Using_Neural.
11. Tom Fawcett, «An Introduction to ROC Analysis». Pattern Recognition Letters. 27 (8): 861–874. DOI:10.1016/j.patrec.2005.10.010, 2006
12. Miron B. Kursa, Witold R. Rudnicki. Feature Selection with the Boruta Package. Journal of Statistical Software, 36(11) , p. 1–13, 2010.
13. S.I. Dimitriadis, Dimitris Liparas, Magda N. Tsolaki, «Random forest feature selection, fusion and ensemble strategy: Combining multiple morphological MRI measures to discriminate among healhy elderly, MCI, cMCI and alzheimer’s disease patients: From the alzheimer’s disease neuroimaging initiative (ADNI) database» Journal of Neuroscience Methods, Volume 302, 2018, Pages 14-23.
14. S. Xuan, G. Liu, Z. Li, L. Zheng, S. Wang and C. Jiang, «Random forest for credit card fraud detection,» 2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC), Zhuhai, 2018, pp. 1-6.
15. Sanaz Nami, Mehdi Shajari, Cost-sensitive payment card fraud detection based on dynamic random forest and k-nearest neighbors, Expert Systems with Applications, Volume 110, 2018, pp 381-392.
16. Eshghi A, Kargari M. Detecting frauds using customer behavior trend analysis and known scenarios. IJIEPR. 2018, pp 91-101.
17. Deshpande D., Deshpande S., Thakare V. Analysis of Online Suspicious Behavior Patterns. Ambient Communications and Computer Systems. Advances in Intelligent Systems and Computing, vol 696. Springer, Singapore, 2018, pp 485-495
18. Gábor Szirtes, Javier Orozco, István Petrás, Dániel Szolgay, Ákos Utasi, Jeffrey F. Cohn, Behavioral cues help predict impact of advertising on future sales, Image and Vision Computing, Volume 65, 2017, pp 49-57