Инженерный вестник Дона, №11 (2024)
ivdon.ru/ru/magazine/archive/n11y2024/9594

Обнаружение нелегитимных вычислительных процессов в АРМ с использованием машинного обучения

Р.Р. Богдалов, А.С. Большаков

Московский технический университет связи и информатики

Аннотация: Защита конечных точек информационной системы от кибератак обусловливает поиск и развитие методов выявления таких атак с использованием искусственного интеллекта. Динамика нарастания количества информационных угроз различного типа приводит к необходимости применения методов машинного обучения для классификации функционирования АРМ, в том числе вычислительных процессов в АРМ. **Цель исследования**: классификация вычислительных процессов созданной базы данных для обнаружения нелегитимных процессов с учетом минимизации количества параметров процессов для достижения приемлемого качества обнаружения. **Методы:** в качестве математического аппарата предлагается использовать модель, обученную на созданном датасете, и корреляционную матрицу на основе коэффициентов Пирсона для определения группы параметров вычислительных процессов. **Результаты:** проведен анализ набора данных на основе коэффициентов корреляции Пирсона, позволяющий минимизировать количество параметров входных данных модели. **Предложено** использовать метод случайного леса для функционирования модели при решении задачи бинарной классификации обнаружении нелегитимных вычислительных процессов в АРМ. Эффективность предложенной модели оценивается метриками классификации: Precision, Recall, **Проведено** тестирование разработанной модели при фиксированных объемах, обучающей и тестирующей выборок. Проведена оценка работы модели с помощью ROC-кривой и PR-кривой.

Ключевые слова: машинное обучение, бинарная классификация, вычислительные процессы, база данных, обработка данных, тестирование модели.

Актуальность защиты конечных точек информационных систем не вызывает сомнений. С использованием методов искусственного интеллекта появилась возможность обнаруживать неизвестные информационные угрозы, возникающие с применением различных техник и подтехник, согласно матрице ATT&CK [1] на стадиях реализации атак. В работах [2, 3] уделено внимание обнаружению атак с использованием методов машинного обучения на внешнем периметре информационной системы, в работе [4] рассматривается возможность выявления вредоносной активности по состоянию поведения системных вызовов операционной системы на АРМ. В

данной работе предложен способ определения вредоносной активности путем анализа вычислительных процессов, функционирующих в АРМ.

Анализ вычислительных процессов важен при обеспечении защиты информации. Множество программ по защите информации используют анализ процессов для повышения безопасности системы. Например, антивирус, Endpoint Detection and Response, Security information and event management, Data Leak Prevention. Вычислительные процессы имеют большое количество параметров, проанализировав которые, можно получить информацию о выполняемом процессе с точки зрения его легитимности, разделив их на «хорошие» и «плохие».

Однако, анализ вычислительных процессов распространен в информационной защите, но открытых баз данных процессов в сети нет. Поэтому была создана собственная база данных вычислительных процессов для машинного обучения, диаграмма, поясняющая структуру формирования которой представлена на рис. 1.

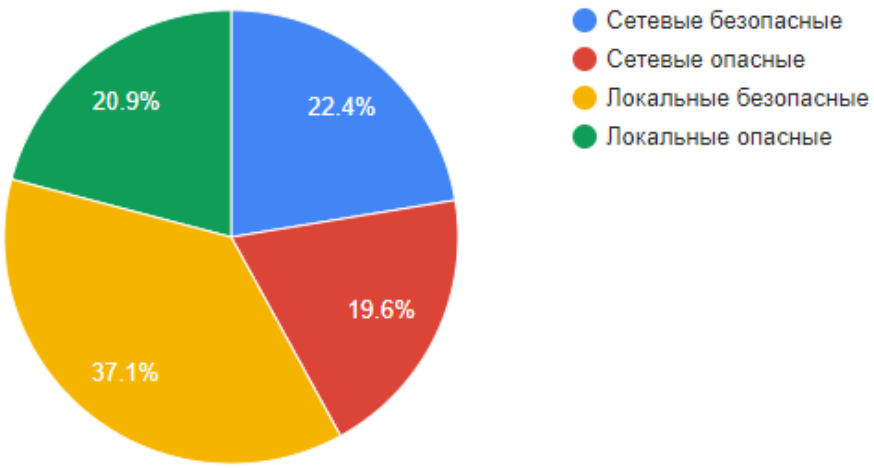

Рис. 1. – Диаграмма структуры базы данных вычислительных процессов

На рис. 1 отображены доли процессов, работающих на компьютере:

- локально, это различные редакторы или приложения,

- сетевые процессы, которые имеют доступ непосредственно к сети интернет.

Для их формирования были созданы 36 скриптов на языке python, конвертированные в файлы формата exe. Среди них присутствовали программы, которые нагружали систему в общем и по отдельным параметрам, таким как GPU, CPU, Сеть и т.д. Помимо локальных действий некоторые программы были нацелены на сеть, перегружая ее, либо имитировали кражу данных или запись действий на компьютере.

Создавать скрипты для «хороших» процессов не было необходимости, так как в качестве таких процессов выступали процессы компьютера, которые имелись до создания скриптов.

Каждый процесс имеет более 50 различных параметров [5]. В качестве сканера процессов компьютера выступала программа Process Explorer [6], созданная для сканирования процессов на компьютерах с операционной системой Windows. На рис. 2 приведен скан интерфейса данной программы.

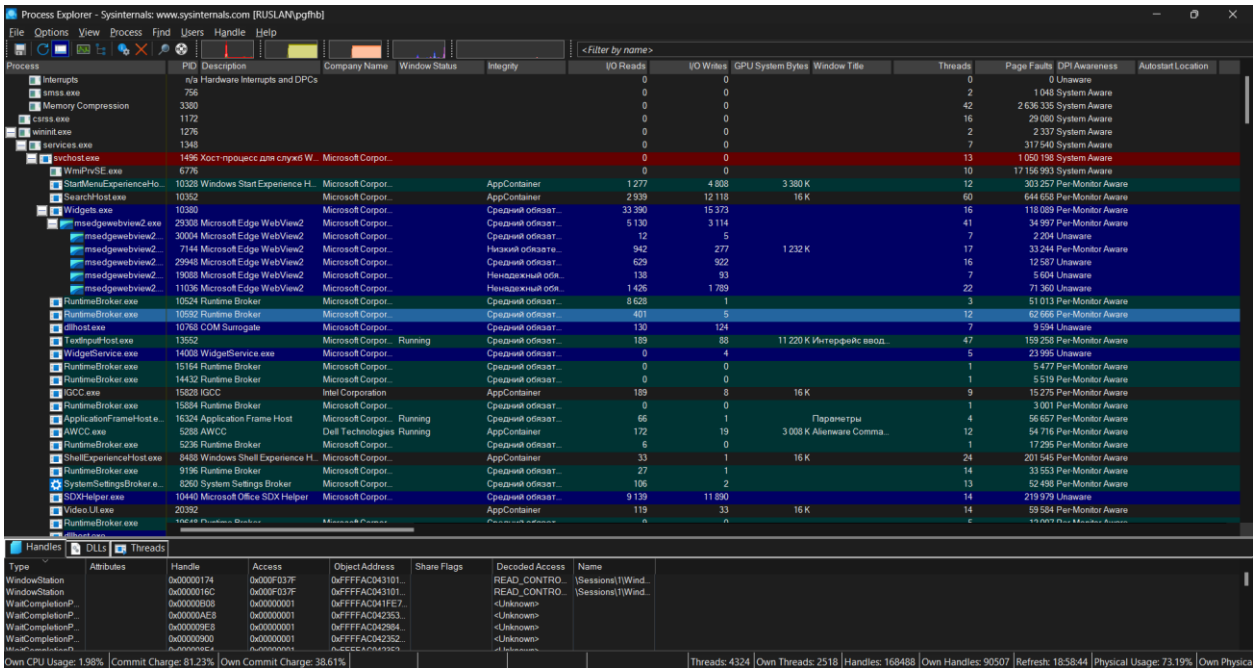

Рис. 2. – Визуализация интерфейса Process Explorer

После запуска сканер собирал всю информацию о процессах, которая записывалась в базу данных. Утилита Process Explorer позволила самостоятельно выбирать параметры, которые отображались при сканировании. На рис. 3 представлена панель с выбором параметров процессов.

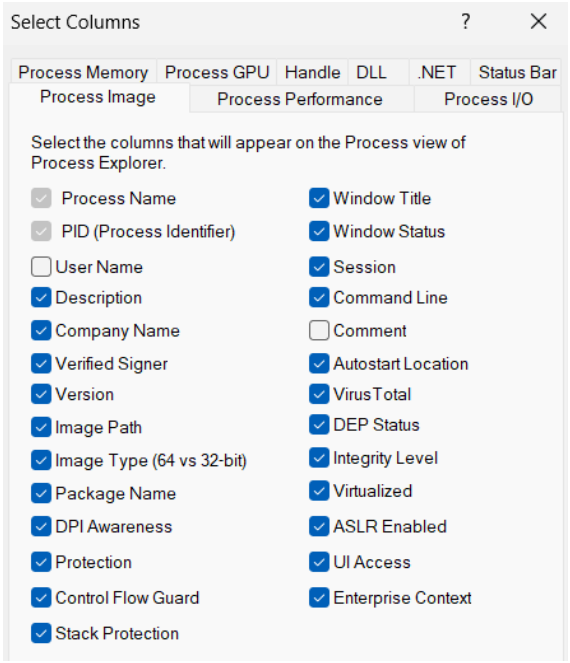

Рис. 3. – Панель выбора процессов

После 50 циклов сбора информации все данные были загружены в единую таблицу. В качестве базы данных был использован PostgreSQL. На рис. 4 можно увидеть фрагмент полученной таблицы, формат которой был использован при создании базы данных вычислительных процессов.

В сформированную базу данных входит 150 тысяч записей, причем каждая запись содержит 61 параметр, наименования которых представлены на рис. 4.

С точки зрения достижения быстродействия принятия решения об обнаружении нелегитимного вычислительного процесса целесообразно

произвести обработку содержимого базы данных с целью сокращения параметров путем создания корреляционной матрицы всех записей.

	tion	priority bigint	handles bigint	start_time text	context_switches bigint	cycles bigint	io_other bigint	io_read_bytes bigint	io_write_bytes bigint	io_other_bytes bigint
952		8	158	15:52:29 01.04.2024	1559636	73602753778	0	0	0	0
953		8	114	19:53:23 17.04.2024	11	33729966	0	0	0	0
954		8	678	11:47:03 12.04.2024	1094877	711500836642	1162712	270000	16000	343000
955		8	520	15:50:19 01.04.2024	1263399	3404809645088	0	0	0	0
956		8	1602	1:27:31 16.04.2024	185739	27149943550	91394	583000000	225000000	36000000
957		8	998	19:30:34 17.04.2024	1541987	121170549020	18405	453000000	163000000	3221000
958		8	1587	22:44:29 16.04.2024	53506	30715626628	11702	232000000	201000000	21000000
959		10	868	19:22:24 17.04.2024	599827	99672464790	4627	2193000000	119000000	2239000
960		8	429	18:54:43 14.04.2024	1018	1627520696	2572	2060000	267000	851000
961		8	469	11:47:03 12.04.2024	36796	12031931106	1433	26000	8	180000
962		8	194	19:52:30 17.04.2024	460	189899696	531	15000	0	283000
963		8	316	15:50:19 01.04.2024	342	6842487648	0	0	0	0
964		8	2080	15:50:18 01.04.2024	97631	634152131118	0	0	0	0
965		8	77	19:53:07 17.04.2024	105	490005174	147	54000000	148000000	11000
966		8	279	11:46:07 12.04.2024	2322	7636836032	200	0	0	12000
967		8	1067	11:46:03 12.04.2024	213526	195849584880	505364	10000000	79000	88000000
968		8	722	19:22:47 17.04.2024	2749	2236267296	8145	436000	168	8224000
969		8	401	11:46:05 12.04.2024	3373	2341643468	4006	72000000	76000000	873000
970		8	525	15:50:19 01.04.2024	4078	23543318762	0	0	0	0

Рис. 4. – Фрагмент таблицы в PostgreSQL

Такая матрица корреляций позволила выявить связи между параметрами, как положительные, так и отрицательные. С точки зрения машинного обучения – это важно по нескольким причинам:

1. Для обучения нужны независимые параметры
2. Корреляционные связи позволяют выявить переизбыток параметров
3. Избыток параметров может привести к переобучению модели

Для того, чтобы избежать корреляций была построена матрица корреляций, которая показывает все зависимости между параметрами. На рис. 5 представлена матрицы корреляций.

Красным обозначены положительные корреляции, синим – отрицательные и белым – неинформативные.

После анализа корреляционных связей были исключены те параметры, которые имели зависимости друг с другом выше 0,7. На рис. 6 можно увидеть матрицу корреляций после исключения коррелирующих параметров.

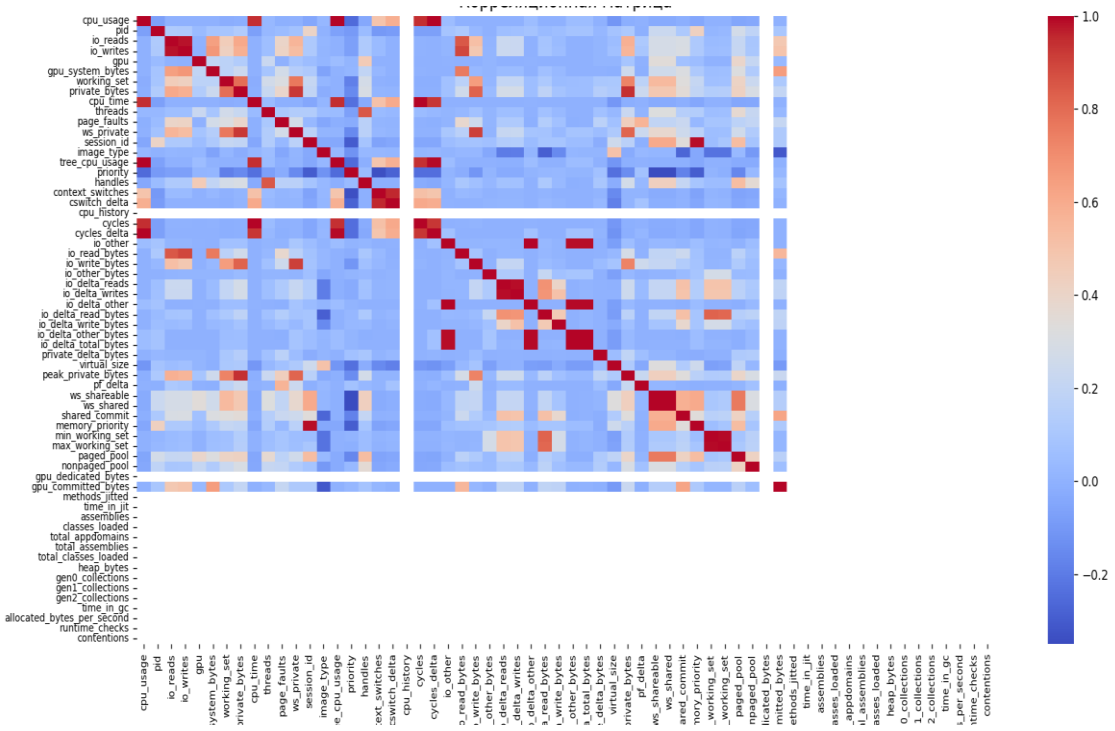

Рис. 5. – Корреляционная матрица данных о процессах на компьютере

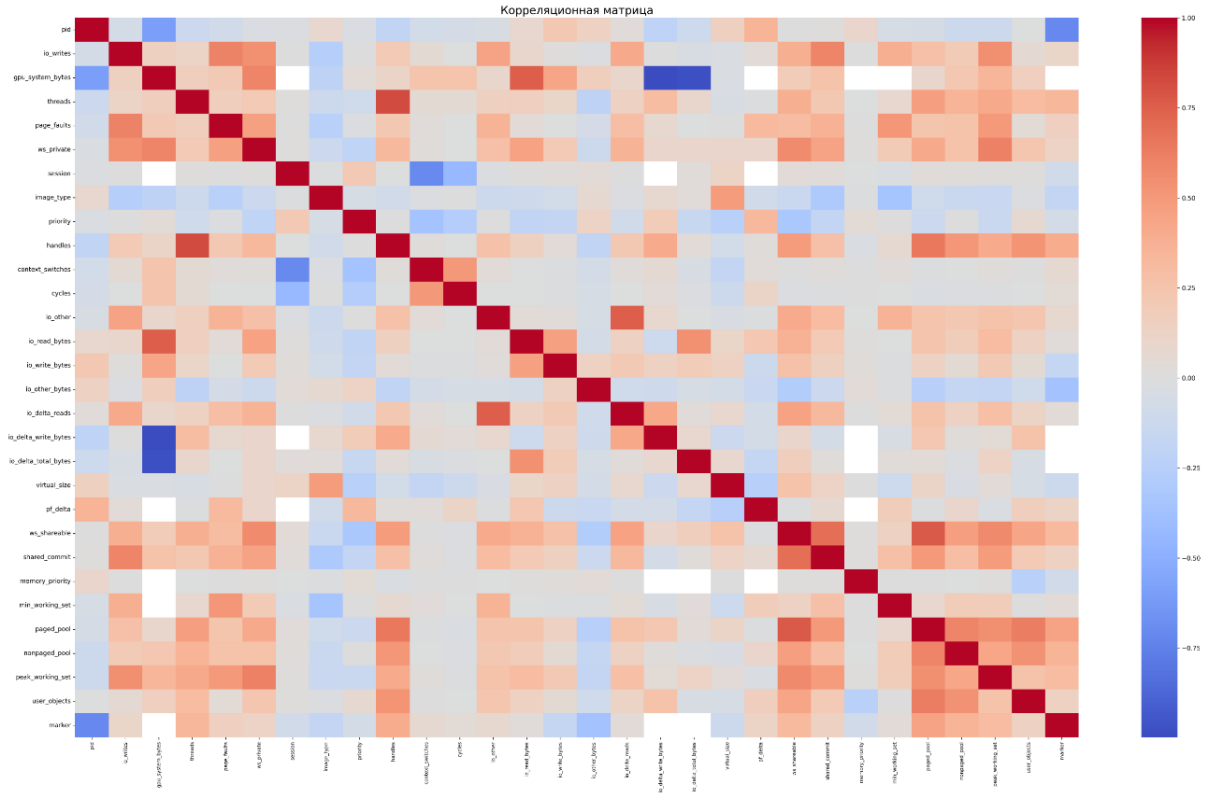

Рис. 6. – Итоговая матрица корреляций

Приведенная на Рис. 6 матрица предполагает исключение:

1. Параметров с положительной корреляцией более 0,7 между собой. Это можно судить по отсутствию красных квадратов, кроме основной диагонали;
2. Параметров с отрицательной корреляцией более 0,7 между собой. Это можно судить по отсутствию синих квадратов;
3. Неинформативных параметров.

В результате количество параметров сократилось до 30 согласно корреляционной связи, не превышающей численное значение равное 0,7.

После обработки базы данных была добавлена метка в качестве дополнительного параметра. Метка имеет только два значения «1» и «0». Она необходима для классификации процесса как легитимного – 1 или нелегитимного – 0.

В итоге получена база данных процессов, которая использовалась в машинном обучении. Обучение модели обнаружения нелегитимных процессов проводилось на 80% записей созданной базы данных, 20% записей базы данных отводилось для тестирования оценки пригодности модели. Этот подход позволяет модели изучать закономерности на основе обучающих данных, а затем оценивать ее производительность на невидимых тестовых данных, обеспечивая понимание ее возможностей обобщения. В данной работе для решения задачи классификации использовался алгоритм случайного леса. Этот алгоритм представляет собой ансамблевый метод, который строит несколько деревьев решений, а затем объединяет их прогнозы в процессе голосования для принятия окончательного решения [7,8,9].

Для оценки качества обученной модели были построены графики ROC кривой и PR кривой, приведенные на рис. 7.

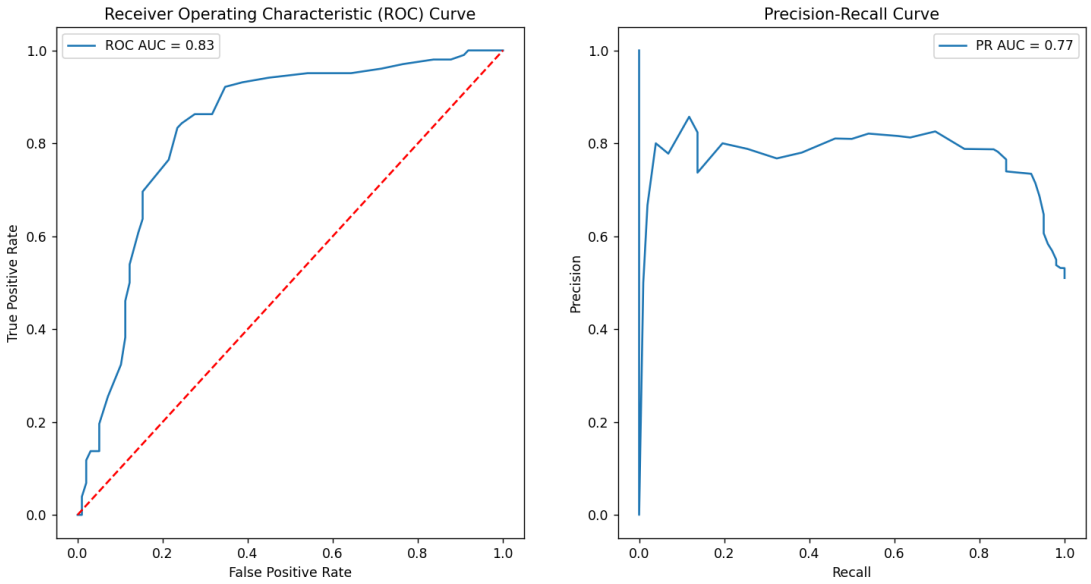

Рис. 7. – Графики метрик обученной модели

Зависимости ROC и PR-кривые позволяют оценить качество классификации и сделать выводы о способности модели различать классы, особенно в условиях несбалансированных данных.

ROC-кривая дает информацию о производительности модели классификации, демонстрируя, насколько хорошо она может различать положительные и отрицательные классы. График ROC демонстрирует кривую, которая постоянно превышает диагональную линию, что указывает на сильную способность классификатора различать классы. При значении AUC 0,83 модель демонстрирует высокий уровень диагностической точности [10].

Кривые точности положительных прогнозов и полноты (способности классификатора идентифицировать все фактические положительные прогнозы) иллюстрируют их взаимосвязь. Анализируя кривую точности и полноты, можно получить представление о компромиссе между точностью и полнотой в модели классификации. Отображаемая кривая демонстрирует постоянный уровень точности по мере увеличения показателя полноты до достижения определенного порога, после чего происходит значительное

снижение точности. Эта закономерность предполагает, что по мере того, как порог выявления положительных случаев становится более строгим, классификатору становится все труднее их правильно классифицировать. Значение PR AUC, равное 0,77, предполагает, что классификатор хорошо справляется с задачей, даже когда сталкивается с потенциальным дисбалансом в распределении классов [11].

Обученная модель может быть применена несколькими способами:

1. Использовать модель в фоновом режиме. То есть сканирование будет происходить без остановки пока работает компьютер.
2. Использовать модель как разовый сканер. То есть запускать сканирование вручную по желанию пользователя.
3. Интегрировать модель в существующие методы программной защиты конечных точек АРМ, в том числе в комплексе с другими средствами обнаружения киберугроз.

Таким образом, предложен подход для классификации вычислительных процессов на автоматизированном рабочем месте с использованием методов машинного обучения. В ходе исследования создана база данных процессов, проведен их сбор, обработка и анализ для дальнейшего обучения модели. Анализ матрицы корреляцией параметров позволил определить эффективное количество параметров для достижения эффективной точности обнаружения вредоносных процессов. Применение алгоритма случайного леса обеспечило приемлемые результаты классификации, что подтверждается анализом ROC и PR-кривых. Предложенная модель может применяться для мониторинга вычислительных процессов, обнаружения аномалий и интеграции в системы защиты информации.

Литература

1. ATT&CK. 2023. URL: attack.mitre.org/datasources/DS0009/.
2. Большаков А.С., Хусаинов Р.В., Осин А.В. Обнаружение аномалий трафика с использование нейронной сети для обеспечения защиты информации. I-METHODS. 2021. №4. Сс. 1-12.
3. Большаков А.С., Губанкова Е.В. Обнаружение аномалий в компьютерных сетях с использованием методов машинного обучения. REDS: Телекоммуникационные устройства и системы. 2020. №1. С. 37-43.
4. Зайченко И.А., Большаков А.С. Об использовании системных вызовов WIN-API для обнаружения модифицированного вредоносного ПО. Телекоммуникации и информационные технологии. 2022. №2. Сс. 28-36.
5. О процессах и потоках. 2023. URL: learn.microsoft.com/ru-ru/windows/win32/procthread/about-processes-and-threads.
6. Процессы, потоки и задания Windows. 2021. URL: sysadminium.ru/kak_ustroen_windows-processy_windows/#Processy.
7. Breiman Leo. Random Forests. Statistics Department University of California Berkeley. 2001. 33 с.
8. Шелухин О.И., Ерохин С.Д., Полковников М.В. Технологии машинного обучения в сетевой безопасности, под ред. доктора технических наук О.И. Шелухина. Горячая линия – Телеком, 2021. 360 с.
9. Шелухин О.И., Зегжда Д.П., Раковский Д.И. Интеллектуальные технологии информационной безопасности, Учебное пособие для вузов. Горячая линия. – Телеком, 2023. 384 с.
10. Pérez-Fernández Sonia, Martínez-Camblor Pablo, Filzmoser Peter and Corral Norberto. nsROC: An R package for Non-Standard ROC Curve Analysis. The R Journal. 2018. №10. 22p.

11. Precision-Recall Curve | ML. 2024. URL: geeksforgeeks.org/precision-recall-curve-ml/.

References

1. ATT&CK. 2023. URL: attack.mitre.org/datasources/DS0009/.
2. Bol'shakov A.S., Husainov R.V., Osin A.V. I-METHODS. 2021. №4. pp. 1-12.
3. Bol'shakov A.S., Gubankova E.V. REDS: Telekommunikacionnye ustrojstva i sistemy. 2020. №1. pp. 37-43.
4. Zajchenko I.A., Bol'shakov A.S. Telekommunikacii i informacionnye tehnologii. 2022. №2. pp. 28-36.
5. O processah i potokah [About processes and threads]. 2023. URL: learn.microsoft.com/ru-ru/windows/win32/procthread/about-processes-and-threads.
6. Processy, potoki i zadanija Windows [Windows Processes, Threads, and Jobs]. 2021. URL: sysadminium.ru/kak_ustroen_windows-processy_windows/#Processy.
7. Breiman Leo. Random Forests. Statistics Department University of California Berkeley. 2001. 33 p.
8. Sheluhin O.I., Erohin S.D., Polkovnikov M.V. Tehnologii mashinnogo obuchenija v setevoj bezopasnosti, pod red. doktora tehnicheskih nauk O.I. Sheluhina [Machine learning technologies in network security, edited by O.I. Shelukhin, Doctor of Technical Sciences]. Gorjachaja linija. Telekom, 2021. 360 p.
9. Sheluhin O.I., Zegzhda D.P., Rakovskij D.I. Intellektual'nye tehnologii informacionnoj bezopasnosti, Uchebnoe posobie dlja vuzov [Intelligent information security technologies, A textbook for universities]. Gorjachaja linija. Telekom, 2023. 384 p.

10. Pérez-Fernández Sonia, Martínez-Camblor Pablo, Filzmoser Peter and Corral Norberto. nsROC: An R package for Non-Standard ROC Curve Analysis. The R Journal. 2018. №10. 22p.
11. Precision-Recall Curve | ML. 2024. URL: geeksforgeeks.org/precision-recall-curve-ml/.

Дата поступления: 27.09.2024

Дата публикации: 14.11.2024