

УДК 004.056

doi: 10.15622/rcai.2025.097

ОБНАРУЖЕНИЕ АНОМАЛИЙ В КОНТЕЙНЕРНЫХ СИСТЕМАХ НА ОСНОВЕ ПРОФИЛИРОВАНИЯ И НЕЙРОННОЙ СЕТИ AE-LSTM¹

И.В. Котенко (*ivkote@comsec.spb.ru*)

М.В. Мельник (*mkmxvh@gmail.com*)

Санкт-Петербургский Федеральный исследовательский центр РАН,
Санкт-Петербург

Особенности архитектуры контейнерных систем создают дополнительные риски, позволяя злоумышленникам негативно воздействовать не только на целевой контейнер, но и на другие контейнеры, размещенные на том же хосте, а также на саму операционную систему. Для повышения точности обнаружения аномалий в контейнерных системах предложена методика создания программных систем обнаружения аномального поведения за счёт оценки состояния элементов контейнерных систем. Данная методика основана на создании эталонных профилей легитимных значений параметров элементов контейнерных систем и использовании нейронной сети AE-LSTM. В качестве таких параметров могут выступать точки монтирования, сетевые интерфейсы, ограничения на использование вычислительных ресурсов и другие. Обученная на эталонном профиле модель позволяет выполнить классификацию текущего состояния контейнера как легитимного или аномального на основе вычисления ошибки реконструкции данных. Если ошибка реконструкции превышает заданный предел – регистрируется аномалия. Точность представленного составляет от 96,9% до 98,7% в зависимости от модели и элемента контейнерной системы.

Ключевые слова: обнаружение аномалий, контейнерные системы, элементы контейнерной системы, нейронные сети.

¹ Исследование выполнено за счет гранта Санкт-Петербургского научного фонда № 23-РБ-01-09.

Введение

Контейнеры, как и виртуальные операционные системы (ОС), представляют собой динамические среды выполнения программного кода. В последнее время они становятся основополагающим элементом при развертывании различных систем, особенно в условиях, когда оптимизация вычислительных ресурсов играет решающее значение. Эффективность контейнерных систем достигается за счет интеграции только необходимых программных компонентов, что является ключевым фактором при создании и развертывании сложных высоконагруженных систем.

Помимо классических проблем безопасности (уязвимости, неправильные конфигурации и другие), контейнеры сталкиваются с рядом специфических проблем. Такие проблемы возникают из-за их архитектурных особенностей, обуславливаемых реализацией основного принципа контейнеризации – WORA (Written Once And Run Anywhere, написано один раз и работает везде) [Dakić, 2024], который позволяет контейнеру использовать ядро хостовой ОС.

Атаки могут быть обнаружены благодаря подходу на основе профилирования, поскольку они часто сопровождаются необычным поведением, которое не свойственно при штатном функционировании наблюдаемой системы. Однако, используемые профили могут содержать большие объемы информации, особенно если в качестве данных используются системные вызовы (System Calls, Syscalls). Кроме того, для реализации подхода на основе профилирования с помощью сложных методов глубокого машинного обучения, например, таких, как управляемые рекуррентные блоки (Gated Recurrent Unit, GRU) или долгая краткосрочная память (Long Short-Term Memory, LSTM), могут потребоваться значительные объемы вычислительных ресурсов и времени. В то же время может оказаться недостаточно использования простых методов, таких, как автокодировщики (Autoencoder, AE) или перцептроны (Perceptron, P).

Для повышения точности обнаружения аномального поведения и снижения затрат вычислительных ресурсов, в статье представлена методика реализации программных систем обнаружения аномального поведения контейнерных систем. Предлагаемая методика предполагает создание эталонных профилей легитимных значений параметров элементов контейнерных систем и основана на использовании модели нейронной сети AE-LSTM. Архитектура разработанной системы и реализованного прототипа содержит два компонента. Первый компонент – модуль обучения на основе профилей легитимных значений параметров элементов контейнеров, второй – модуль обнаружения.

На этапе обучения создаются эталонные профили, описывающие легитимное поведение элементов контейнерных систем для каждой группы контейнеров. Эти профили состоят из данных о значениях определенных параметров элементов контейнерной системы, собранных за определен-

ные периоды времени. Такие параметры включают сетевые интерфейсы, ограничения на использование вычислительных ресурсов, точки монтирования и другие. После обучения на эталонных данных нейронная сеть способна классифицировать текущее состояние контейнера, определяя его как легитимное или аномальное. В процессе обнаружения аномалий модель анализирует значения параметров на определенном промежутке времени и вычисляет ошибку реконструкции данных. Если ошибка реконструкции этих данных превышает заданное пороговое значение, система сигнализирует о наличии аномалии.

Оценки, полученные в ходе экспериментов, демонстрируют уровень точности показателя Assurance в диапазоне от 96,9% до 98,7% в зависимости от модели нейронной сети AE-LSTM и класса параметров элементов контейнерных систем. Кроме того, представленное решение не требует высоких затрат вычислительных ресурсов (как при обучении модели, так и при обнаружении).

1. Обзор релевантных работ

Многие исследования демонстрируют эффективное применение методов машинного обучения в качестве ключевого метода реализации подхода на основе профилирования в задачах обнаружения аномального поведения [Branitskiy et al., 2017], [Laskov et al., 2004]. В текущих исследованиях используются, в основном, модели глубокого обучения.

В [Kotenko et al., 2024] используется трассировка syscalls для отслеживания поведения в контейнерных системах. После чего выполняется построение гистограмм процессов и передача их в модель нейронной сети AE для обучения и последующего обнаружения аномальных процессов. Авторы отмечают неспособность предложенного подхода к обнаружению определенных типов атак.

В [Tien et al., 2019] представлена система KubAnomaly для обнаружения аномалий на платформе оркестрации Kubernetes. Представленный подход, напротив, основан на разбиении непрерывного потока syscalls на отдельные последовательности. Ввиду большого количества данных авторы приняли решение отслеживать только четыре категории syscalls (файловый ввод-вывод, сетевой ввод-вывод, планировщик и память). Результаты экспериментов демонстрируют точность предложенной модели около 96%.

Подходы, использованные в [Snehi et al., 2021] и [Gantikow et al., 2020], похожи и основаны на использовании модели нейронной сети LSTM и трассировки syscall. В [Snehi et al., 2021] анализируется поведение контейнерной системы на основе syscall. Процесс обнаружения основан на анализе разности предсказанной последовательности syscalls по отношению к действительной. Если значение разности превышает допустимое

пороговое значение, то поведение контейнера считается аномальным. В [Gantikow et al., 2020] представлены два подхода к обнаружению. Первый, как и в [Snehi et al., 2021], основан на трассировке и прогнозировании syscalls, второй – на прогнозировании путей к файлам/каталогам. В целом, можно отметить, что оба представленных подхода хорошо дополняют друг друга.

В [Gupta et al., 2017] используется гибридная нейронная сеть для прогнозирования потребления вычислительных ресурсов во время работы контейнера. Представленная модель состоит из классической модели LSTM и двунаправленной сети долгой краткосрочной памяти (Bidirectional LSTM, BiLSTM). Ряд экспериментов показал, что эта модель демонстрирует хорошую точность, а показатель Recall достигает 94%.

Работы [Castanhel et al., 2021] и [Cui et al., 2021] – схожи и используют общий подход на основе трассировки и техники скользящего окна для сбора и формирования последовательностей syscalls. В [Castanhel et al., 2021] сравнивается эффективность различных методов машинного обучения. В качестве входных данных используются последовательности syscalls, полученные в результате трассировки. Последовательности формируются на основе техники скользящего окна и фильтрации. Авторами отмечается, что точность метода Multilayer Perceptron (MLP) растет вместе с ростом размера скользящего окна и составляет более 89% в [Cui et al., 2021] напротив, используется гибридная нейронная сеть на основе AE-LSTM. В целом, авторы отмечают неплохую точность обнаружения аномалий неконтролируемой моделью нейронной сети.

В [Wang et al., 2023] представлена система DockerWatch на основе двух нейронных сетей CNN и CNN-LSTM, которые используют извлеченный из контейнера исполняемый файл для анализа на предмет вредоносного содержимого.

В результате проведенного анализа релевантных работ можно сделать вывод, что большинство исследований по теме обнаружения аномального поведения направлены на использование syscalls в качестве входных данных для моделей нейронных сетей. Такой подход имеет существенный недостаток, связанный с большими объемами данных, генерируемыми контейнерными системами. Обучение моделей со сложной архитектурой может быть крайне затруднено с точки зрения использования вычислительных ресурсов и времени, а обучение простых моделей может оказаться недостаточным для эффективного и своевременного обнаружения.

2. Подход к обнаружению

Работа контейнерной системы обеспечивается за счет взаимодействия ее ключевых элементов, приложения внутри такой системы и пользователей. В качестве элементов контейнерной системы могут выступать: образ

контейнера (Docker Image), контейнерный движок (Container Engine), контейнерная сеть (Container Networking), хранилище (Volumes), API (Application Programming Interface) и CLI (Command-Line Interface) и механизмы безопасности, определяемые пространством имен (Namespaces), контрольными группами (Control Groups, Cgroups), Seccomp (Secure Computing Mode), AppArmor (Application Armor), SELinux (Security-Enhanced Linux) и возможностями Linux (Capabilities).

Аномалии могут проявляться в изменениях различных значений параметров таких элементов. Например, появление новой точки монтирования может указывать на попытку атаки типа Docker Escape. Запуск контейнера с новым, ранее не использовавшимся образом, может свидетельствовать о внедрении вредоносного программного обеспечения. Изменение IP-адреса (Internet Protocol) контейнера может быть признаком подготовки к атаке “Человек посередине” (Man-in-the-Middle, MitM), IP Spoofing или попытке обойти сетевые политики и правила безопасности. Изменение параметров CapAdd и CapDrop, которые управляют добавлением и удалением Capabilities стандартного набора контейнера, может предполагать попытку повышения привилегий (Privilege Escalation) или обход механизмов безопасности.

Учитывая особенности контейнеров и природу аномалий, предлагаемый подход основывается на создании эталонных профилей, отражающих нормальное поведение элементов этих систем. Каждый профиль формируется для определенной группы контейнеров и некоторого класса параметров на последовательных временных интервалах со смещением в одну минуту. Продолжительность каждого интервала составляет 10 минут.

Таким образом, для каждой категории параметров создается отдельный профиль: для ограничений использования вычислительных ресурсов (RUL), для сетевых интерфейсов (Net), для точек монтирования (Mnt), для образов (Img). Сформированные профили используются для обучения моделей нейронной сети AE-LSTM, которая в дальнейшем применяется для обнаружения аномалий в работе контейнера путем анализа параметров его элементов.

Представленная система включает несколько ключевых компонентов: (1) сбор данных; (2) нормализация данных; (3) обучение и (4) обнаружение.

(1) Сбор данных. Для сбора значений параметров каждого элемента контейнерной системы разработан программный компонент, который в единицу времени (минуту) собирает информацию о сетевых интерфейсах, установленных ограничениях на использование вычислительных ресурсов, точках монтирования и др., как представлено в листинге 1.

Данные о параметрах элементов контейнерной системы

```

1: Net: 127.0.0.1, 172.28.1.10,...
2: RUL: {"CpuShares":0, "Memory":432013312, "KernelMemory":null,...
3: Mnt: /dev/sda2 on /etc/resolv.conf type ext4 (rw,relatime)...
```

В данном листинге в разделе в разделе “Net” представлены сетевые интерфейсы, в разделе “RUL” – установленные лимиты на использование вычислительных ресурсов, “Mnt” – представлены точки монтирования.

(2) Нормализация данных. На первом этапе формируются временные последовательности собранных значений параметров элементов контейнеров. Каждая последовательность охватывает 10-минутный интервал со смещением в 1 минуту. На втором этапе выполняется обработка данных для каждого класса параметров, которая осуществляется с учетом специфики их представления. Например, для профиля Net каждый октет сетевого интерфейса записывается как отдельный элемент в формируемую строку; в результате строка содержит описание всех сетевых интерфейсов. Для профиля RUL на основе предопределённых параметров, таких, как CpuShares, Memory, KernelMemory и др., формируется строка, в которой первое значение соответствует CpuShares, второе – Memory, третье – KernelMemory и так далее; если значение для какого-либо параметра отсутствует, используется значение по умолчанию, равное 0.

(3) Обучение. В основе предлагаемой системы лежит неконтролируемая гибридная модель нейронной сети AE-LSTM, обучение которой осуществляется на основе данных, характеризующих проведение элементов контейнерных систем. Такой подход позволяет избежать необходимости явной разметки данных, которая в реальных условиях затруднена. В результате успешного обучения каждому профилю будет соответствовать модель AE-LSTM, которая представлена на рис. 1.

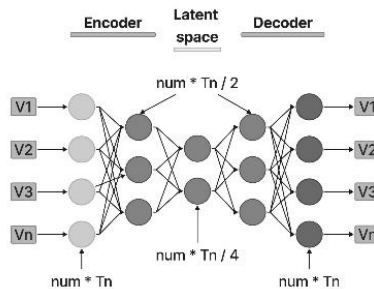


Рис. 1. Модель нейронной сети AE-LSTM

После загрузки данных в модель нейронной сети, кодировщик (Encoder) сжимает входные данные в скрытое представление (Latent space). После чего декодер (Decoder) их восстанавливает из Latent space в исходное состояние. Таким образом, модель обучается минимизировать ошибку реконструкции во время обучения.

Ключевой особенностью данной архитектуры является постепенное уменьшение размерности данных на первом ($\text{num} * T_n / 2$) и втором ($\text{num} * T_n / 4$) уровне скрытого слоя. Размер входного и выходного слоя определяется как количество значений в строке передаваемых данных (num) умноженное на длину временного периода (T_n), которая равна 10). Такая архитектура позволяет предотвращать переобучение модели, сосредотачиваясь на наиболее значимых признаках. В качестве функций активации используются ReLU (Rectified Linear Unit), которые обеспечивают быстрое вычисление за счёт использования простой функции: если входное значение положительное, то функция возвращает это значение, а если отрицательное или равно 0, то функция возвращает 0.

Кроме того, в ходе ряда экспериментов было установлено, что использование модели AE-LSTM, в отличие от классической модели AE, обеспечивает значительно более высокие показатели точности.

При обучении модели нейронной сети использовалась комбинация алгоритма оптимизации ADAM (Adaptive Moment Estimation) и функции потерь MSE (Mean Squared Error). ADAM быстро и эффективно настраивает параметры модели, такие как веса (Weights) и смещения (Biases). Это позволяет автоматически подстраивать скорость обучения для каждого из этих параметров. Функция MSE, измеряя средний квадрат разности между предсказанными и фактическими значениями, обеспечивает плавные градиенты, что упрощает и стабилизирует процесс обучения, приводя к более точным прогнозам.

(4) Обнаружение. На этапе обнаружения данные за 10-минутный интервал передаются в соответствующую модель для классификации состояния элемента контейнера как легитимного или аномального. Для поступивших данных вычисляется MSE между исходными и реконструированными данными. Если это значение превышает установленное пороговое значение, данные классифицируются как аномальные.

Пороговое значение устанавливается как среднее значение MSE плюс коэффициент, умноженный на стандартное отклонение. Среднее и стандартное отклонение MSE рассчитывается на основе вычисления MSE реконструкции каждого примера из набора данных, представляющих нормальное поведение. Это обеспечивает гибкую настройку чувствительности, балансируя между точностью и ложными срабатываниями.

3. Эксперименты

Для подтверждения эффективности разработанной методики и программного прототипа был создан испытательный стенд, подготовлены наборы данных, обучены модели нейронной сети и выполнены оценки точности обученных моделей.

Испытательный стенд. Испытательный стенд включает две виртуальные машины под управлением Ubuntu 22.04.5 LTS. Первая виртуальная машина содержит программный компонент для сбора и передачи значений параметров элементов контейнеров в модель нейронной сети, а также два Docker-контейнера: RDP-сервер (Remote Desktop Protocol) и SSH-сервер (Secure Shell). Вторая виртуальная машина также работает под управлением Ubuntu 22.04.5 LTS и используется для взаимодействия с Docker-контейнерами посредством RDP и SSH. Это взаимодействие включает в себя запуск утилит, выполнение команд ОС, обновление параметров элементов контейнерных систем и другие.

Наборы данных. Всего сформировано четыре набора данных, которые получены в результате работы двух контейнеров.

Два набора данных (rdp-n.txt и ssh-n.txt) включают данные легитимных параметров элементов контейнерных систем, на которых были обучены модели нейронной сети AE-LSTM.

Набор данных rdp-n.txt содержит: редактирование сетевых интерфейсов с использованием легитимных значений, монтирование диска внутри контейнера (легитимные пути целевых устройств – /dev/sda2/ и /dev/sda3/ и другие, легитимные пути назначения – /mnt/share_1, /mnt/share_2, /mnt/share_3 и другие).

Набор данных ssh-n.txt содержит: изменение ограничений на использование вычислительных ресурсов (cpuset-cpus, memory, memory-swap, cpu-shares, cpu-quota, cpu-period, memory-reservation) на легитимные значения; запуск контейнеров на основе проверенных образов.

Два остальных набора данных (rdp-a.txt и ssh-a.txt) включают как данные легитимных значений параметров, так и не легитимных.

Набор данных rdp-a.txt содержит: редактирование сетевых интерфейсов с использованием легитимных и не легитимных значений, монтирование диска внутри контейнера (легитимные пути целевых устройств – /dev/sda2/ и /dev/sda3/ и другие, легитимные пути назначения – /mnt/share_1, /mnt/share_2, /mnt/share_3 и другие, нелегитимные пути целевых устройств – /dev/sdb2/ и /dev/sdb3/ и другие, нелегитимные пути назначения – /mnt/escape_1, /mnt/escape_2, /mnt/escape_3 и другие).

Набор данных ssh-a.txt содержит: изменение ограничений на использование вычислительных ресурсов (cpuset-cpus, memory, memory-swap, cpu-shares, cpu-quota, cpu-period, memory-reservation на легитимные и не легитимные значения), запуск контейнеров на основе проверенных и не проверенных образов.

Обучение. Обучение моделей проводилось на виртуальной машине, которая включает следующие характеристики: ОС: Ubuntu 22.04.5 LTS; RAM (Random Access Memory): 8 GB (Gigabyte); CPU (Central Processing Unit): AMD Ryzen 5 5600X 2 Virtual Core Processor 3.70 GHz. Время для обучения каждой модели составило не более 1 минуты.

В ходе ряда экспериментов было установлено, что при использовании представленной архитектуры наилучшие показатели обнаружения модели достигаются при уровне потерь в диапазоне от 0,5% до 1%. Такой уровень потерь достигается при установленных параметрах batch size (количество примеров, используемых в одной итерации обучения), равном 1, и epoch (количество итераций обучения), равном 5.

В результате были обучены четыре модели нейронной сети AE-LSTM. Данные для каждой модели были агрегированы из двух наборов данных (rdp-n.txt и ssh-n.txt), а затем с помощью фильтрации преобразованы для обучения каждой модели в соответствии с типом параметров элементов контейнеров. Таким образом, модель Net содержит данные о легитимных сетевых интерфейсах, RUL – данные о легитимных значениях ограничения вычислительных ресурсов, MNT – данные о легитимных точках мониторинга, Img – данные о легитимных образах контейнеров. Кроме того, наборы данных были оптимизированы путем удаления дублирующихся последовательностей, в результате чего из нескольких идентичных последовательностей оставалась только одна.

Результаты обнаружения. Оценка эффективности разработанного прототипа проводилась с помощью меры Accuracy на наборах данных: rdp-n.txt, ssh-n.txt, rdp-a.txt и ssh-a.txt). Результаты точности разработанных моделей для модели net составляют 98,9 %, для rul – 96,9 %, mnt – 98,7 %, img – 97,2 %.

4. Анализ результатов

В данной статье представлена методика разработки программных систем обнаружения аномального поведения в контейнерах, которая основана на оценке состояния их элементов. Ключевой особенностью и научной новизной предлагаемого решения является использование значений параметров элементов контейнерной системы на временных промежутках в качестве входных данных для моделей нейронной сети AE-LSTM.

В отличие от предыдущих исследований [Kotenko et al., 2024], [Tien et al., 2019], [Snehi et al., 2021], [Gantikow et al., 2020], которые используют syscalls в качестве входных данных для моделей нейронных сетей, предлагаемое решение использует данные о параметрах элементов контейнерных систем. Вследствие чего нагрузка на вычислительные ресурсы значительно ниже, чем при использовании syscalls.

Следует отметить, что работы [Castanhel et al., 2021], [Cui et al., 2021], в которых используется техника скользящего окна для формирования последовательностей syscalls, ввиду большого потока данных, генерируемых динамичными контейнерами, могут столкнуться со сложностями при обработке данных. В отличие от этого подхода, метод анализа параметров элементов контейнерных систем не сталкивается с данной проблемой, поскольку объем таких данных невелик, а их анализ осуществляется каждую минуту. При необходимости интервал между итерациями анализа можно сократить, а длину временной последовательности увеличить.

В [Gupta et al., 2017] представлен подход на основе прогнозирования показателей нагрузки на вычислительные ресурсы во время работы контейнера. Применённый подход не сможет обеспечить обнаружение аномалий, природа которых не связана с изменением нагрузки на вычислительные ресурсы, например, запуск контейнера с установленными ограничениями на использование вычислительных ресурсов, изменение сетевых интерфейсов, монтирование диска хостовой операционной системы из контейнера. В отличие от этого подхода, предложенный в данном исследовании подход, напротив, сможет обеспечить обнаружение таких аномалий.

Однако, у предложенного решения есть ряд недостатков. Ввиду того, что учитываются только данные о параметрах элементов контейнера, некоторые атаки не могут быть обнаружены. К таким атакам можно отнести Crypto Mining, Privilege Escalation.

Заключение

В статье представлена методика реализации системы обнаружения аномального поведения в контейнерных системах за счёт оценки состояния элементов контейнерных систем. Предлагаемая методика основана на создании эталонных профилей, отражающих состояние элементов контейнера на временных интервалах.

Новизной и отличительной особенностью предлагаемого подхода к обнаружению аномального поведения в контейнерных системах является анализ временных последовательностей значений параметров элементов контейнерных систем с использованием моделей AE-LSTM.

Результаты проведенного эксперимента показывают, что точность реализованного прототипа составляет от 96,9% до 98,7% в зависимости от модели нейронной сети AE-LSTM и класса параметров элементов контейнерной системы. Следует отметить, что предлагаемый подход нацелен на обнаружение только тех аномалий, природа которых напрямую связана с изменением значений параметров элементов контейнера. Другие аномалии не будут выявлены. Таким образом, данное решение может быть использовано в качестве дополнительного инструмента для обнаружения

аномалий, учитывая, что оно позволяет быстро обучать модели нейронной сети и не требует значительных затрат вычислительных ресурсов для своей работы.

В рамках дальнейших исследований планируется расширить набор анализируемых параметров элементов контейнера с целью повышения точности обнаружения аномалий. В частности, будут исследованы: NetworkMode (режим работы сети), CgroupnsMode (режим пространств имен для групп контроля), IpcMode (режим межпроцессного взаимодействия), CapAdd (возможности, предоставленные контейнеру) и другие параметры, потенциально влияющие на безопасность и стабильность работы контейнеров.

Список литературы

- [Branitskiy et al., 2017] Branitskiy A., Kotenko I. Hybridization of computational intelligence methods for attack detection in computer networks // Journal of Computational Science. – Elsevier, 2017. – No. 23. – P. 145-156. – doi: 10.1016/j.jocs.2016.07.010.
- [Castanhel et al., 2021] Castanhel G.R., Heinrich T., Ceschin F., Maziero C. Taking a peek: An evaluation of anomaly detection using system calls for containers // 2021 IEEE Symposium on Computers and Communications (ISCC), Athens, Greece, 2021. – P. 1-6. – doi: 10.1109/ISCC53001.2021.9631251.
- [Cui et al., 2020] Cui P., Umphress D. Towards unsupervised introspection of containerized application // Proceedings of the 2020 10th International Conference on Communication and Network Security, New York, NY, United States, 2020. – P. 42-51. – doi: 10.1145/3442520.3442530.
- [Dakić, 2024] Dakić P. Software compliance in various industries using ci/cd, dynamic microservices, and containers // Open Computer Science. – 2024. – Vol. 14(1). – P. 20240013. – doi: 10.1515/comp-2024-0013.
- [Gantikow et al., 2020] Gantikow H., Zöhner T., Reich C. Container anomaly detection using neural networks analyzing system calls // 2020 28th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), Västerås, Sweden, 2020. – P. 408-412. – doi: 10.1109/PDP50117.2020.00069.
- [Gupta et al., 2017] Gupta S., Muthiyar N., Kumar S., Nigam A., Dinesh D.A. A supervised deep learning framework for proactive anomaly detection in cloud workloads // 2017 14th IEEE India Council International Conference (INDICON), Roorkee, India, 2017. – P. 1-6. – doi: 10.1109/INDICON.2017.8488109.
- [Kotenko et al., 2024] Kotenko I., Melnik M., Abramenko G. Anomaly detection in container systems: using normal process histograms and an autoencoder // IEEE 25th International Conference of Young Professionals in Electron Devices and Materials (EDM 2024), Altai, Russian Federation, 2024. – P. 1930-1934. – doi: 10.1109/EDM61683.2024.10615118.
- [Laskov et al., 2004] Laskov P., Schafer C., Kotenko I. Intrusion detection in unlabeled data with one-class Support Vector Machines // Lecture Notes in Informatics, No. 46, Dortmund, Germany, July 2004. – P. 71-82.

- [Snehi et al., 2021] Snehi J., Bhandari A., Baggan V., Snehi M., Kaur H. AIDAAS: Incident handling and remediation anomaly-based IDaaS for cloud service providers // 2021 10th International Conference on System Modeling & Advancement in Research Trends (SMART), MORADABAD, India, 2021. – P. 356-360. – doi: 10.1109/SMART52563.2021.9676296.
- [Tien et al., 2019] Tien C.W., Huang T.Y., Tien C.W., Huang T.C., Kuo S.Y. KubAnomaly: Anomaly detection for the Docker orchestration platform with neural network approaches // Engineering reports. – 2019. – Vol. 1(5). – P. e12080. – doi: 10.1002/eng2.12080.
- [Wang et al., 2023] Wang Y., Wang Q., Qin X., Chen X., Xin B., Yang R. DockerWatch: a two-phase hybrid detection of malware using various static features in container cloud // Soft Computing. – 2023. – Vol. 27(2). – P. 1015-1031. – doi: 10.1007/s00500-022-07546-2.