Детектирование сетевых распределенных атак на основе алгоритмов машинного обучения

Detection of network distributed attacks based on machine learning algorithms

Пиховкин / Pikhovkin N.
Николай Леонидович
(n.pikhovkin@gmail.com)
ФГАОУ ВО «Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики»
(Университет ИТМО), аспирант.
г. Санкт-Петербург

Попов / Popov I.
Илья Юрьевич
(ilyapopov27@gmail.com)
Университет ИТМО, аспирант.
г. Санкт-Петербург

Карманов / Karmanov A.
Андрей Геннадиевич
(karmanov.nip@gmail.com)
кандидат технических наук, доцент.
Университет ИТМО, доцент ГИС.
г. Санкт-Петербург

Карманова / Karmanova N.
Наталия Андреевна
(natali.karmanova@gmail.com)
Университет ИТМО, лаборант.
г. Санкт-Петербург

Ключевые слова: информационная безопасность – IT security; сетевые атаки – network attacks; машинное обучение – machine learning; анализ данных – data analysis.

В статье описывается метод детектирования атак, включающий в себя генерацию наборов вредоносного трафика и трафика легитимных пользователей, а также проводится оценка возможности применения алгоритмов классификации на основе машинного обучения с учителем для детектирования сетевых распределенных атак типа «отказ в обслуживании».

The article describes attack-detection method including generation of malicious traffic and legitimate users traffic patterns and assesses the possibility of using classification algorithms based on machine learning with a tutor targeted at detection of service denial-type distributed attacks.

Введение

В настоящее время наблюдается рост количества распределенных атак на глобальные компьютерные сети, значительная часть из которых приходится на атаки типа "Отказ в обслуживании" (DDoS). По данным российской компании Qrator Labs, уже сейчас проводятся реальные DDoS-атаки до 1 Тбит/с, значительно увеличивается количество сложных атак на протоколы прикладного уровня [7, 9]. DDoS-атаки влияют на один из основополагающих принципов информационной безопасности – доступность информационных ресурсов [22]. В каждом конкретном случае DDoS может либо непосредственно причинить вред, либо создать угрозу и потенциальный риск нанесения убытков. Атакам такого рода могут быть подвержены любые сетевые ресурсы, вне зависимости от их владельцев и направления деятельности [10, 19]. Таким образом, одним из наиболее актуальных на сегодняшний день направлений в области информационной безопасности является противодействие DDoS-атакам, в том числе разработка средств защиты, способных справиться со сложными высокоскоростными атаками [2, 21].

Целью работы является рассмотрение существующих алгоритмов классификации данных на основе машинного обучения, проведение оценки их эффективности и разработка метода детектирования сетевых распределенных атак типа "Отказ в обслуживании" с помощью алгоритмов машинного обучения.

В работе рассмотрены существующие алгоритмы классификации данных на основе машинного обучения, проведена оценка их эффективности и разработан метод детектирования сетевых распределенных атак типа "Отказ в обслуживании" с помощью алгоритмов машинного обучения.

Применение машинного обучения в детектировании сетевых атак

Говоря об автоматизации в современных средствах защиты от DDoS-атак, практически всегда подразумевается использование машинного обучения. Если большинство из низкоуровневых типов DDoS-атак можно обнаружить и нейтрализовать при помощи методов "грубой силы", то при защите от атак прикладного уровня невозможно обойтись без применения алгоритмов машинного обучения. При L7-атаках в сторону

защищаемого приложения поступает большое количество сетевых пакетов со множеством атрибутов [20]. Необходимо уметь разделять запросы от ботов и реальных пользователей. В решении этой задачи наиболее эффективным считается использование машинного обучения [4].

Для повышения эффективности процессов обнаружения и противодействия DDoS-атакам было проведено исследование того, как алгоритмы машинного обучения могут быть использованы для противодействия DDoS-атакам при их применении в автоматизированных системах защиты.

Использование механизмов машинного обучения помогает решить проблему обнаружения DDoS-атак как с помощью алгоритмов обучения с учителем, так и без учителя. Первый метод обладает более высокой точностью обнаружения, когда имеются точные и корректные атрибуты атак, но не может обнаруживать неизвестные атаки. Второй, напротив, может помочь в обнаружении неизвестных атак, однако он значительно теряет эффективность при наличии шумов и большого количества ложных данных. Также методы обучения без учителя, как правило, не могут работать в реальном времени, так как требуют более серьезной и продолжительной обработки данных [16].

В данной работе были использованы классификаторы на основе алгоритмов дерева решений, *k*-ближайших соседей и случайного леса из библиотеки машинного обучения *scikit-learn*. Полный перечень программных инструментов, использованных в работе, приведен в табл. 1.

Задача классификации. Задача классификации формулируется следующим образом: на основе обучающей выборки сетевого трафика T_s найти функцию *f(x)* от входных параметров, которая при наименьшем количестве ошибочно классифицированных пакетов предсказывает результирующий класс атаки y для любых новых значений x. Выходные значения берутся из дискретного множества $\{y_1, y_2, \ldots, y_m\}$, включающего все предопределенные значения классов. Иными словами,

$$\exists f : X \to Y,$$

где X – это набор векторов атрибутов сетевых пакетов, Y – набор наименований классов DDoS-атак. Значения целевой зависимости f известны только на объектах конечной обучающей выборки:

$$x^m = \{(x_1, x_2), \ldots, (x_m, y_m)\}.$$

Требуется найти такой алгоритм

$$a : X \to Y,$$

который способен классифицировать произвольный объект $x \in X$. В качестве атрибутов классификации должны использоваться атрибуты пакетов сетевого, транспортного и прикладного уровней модели OSI.

Алгоритм *k*-ближайших соседей. Построение модели классификации на основе алгоритма k ближайших соседей заключается в запоминании обучающей выборки данных. Для того чтобы сделать прогноз для нового экземпляра данных, алгоритм выполняет поиск ближайшей к ней точки обучающей выборки, тем самым находя “ближайших соседей”. Новому экземпляру присваивается метка, принадлежащая ближайшей точке обучающего набора. Алгоритм позволяет рассматривать не одного ближайшего соседа, а их произвольное количество (k), отсюда происходит название алгоритма. Пусть дана обучающая выборка с парами вида «объект-ответ»:

Таблица 1

Программные инструменты, используемые для построения модели классификации

Модуль	Применение
pandas	Первичная обработка данных, работа с размеченными данными, формирование датафреймов. Работает в связке с NumPy, и помимо математических вычислений обеспечивает их агрегацию и визуализацию
sklearn	Отбор признаков, реализация алгоритмов классификации
numpy	Выполнение основных операций над n-массивами и матрицами. Использование механизмов векторизации NumPy повышает производительность и ускоряет выполнение операций над данными
matplotlib	Визуализация данных, формирование графиков и диаграмм

$$x^m - \{(x_1, x_2), \ldots, (x_m, y_m)\}.$$

Пусть на множестве объектов задана функция расстояния $p(x,x')$, которая должна быть достаточно адекватной моделью сходства объектов, то есть чем больше значение этой функции, тем менее схожими являются объекты x,x'. Для произвольного u расположим объекты обучающей выборки x_i в порядке возрастания расстояний до u:

$$p(u, x_{1;u}) \le p(u, x_{2;u}) \le \ldots \le p(u, x_{m;u}),$$

где через $x_{i;u}$ – это объект обучающей выборки, который является i-м соседом объекта u. Аналогично введем обозначение $y_{i;u}$ для ответа на на i-м соседе. Алгоритм ближайших соседей в наиболее общем виде выглядит так:

$$a(u) = argmax \sum_{i=1}^{m} \left[x_{i;u} = y \right] w(i,u),$$

где $w(i,u)$ – это заданная весовая функция, оценивающая степень важности i-го соседа при классификации объекта u. Эта функция должна быть неотрицательной и не возрастать по i-м.

Преимуществом использования алгоритма k-ближайших соседей является легкость интерпретации модели, а также приемлемое качество предсказания, которого можно добиться без применения большого количества настроек. Как правило, построение модели классификации на основе данного алгоритма происходит очень быстро, однако когда размер обучающей выборки становится очень большим, требуется дополнительное время для получения прогноза. Данный алгоритм работает не очень эффективно с наборами данных из сотен и более признаков, и совсем неэффективно в ситуации, когда подавляющее большинство признаков в большей части наблюдений имеют нулевые значения (такой набор данных называется *sparse dataset*, или разреженным). Таким образом, алгоритм k ближайших соседей не так часто используется на практике из-за относительно низкой скорости вычислений и неспособности к обработке большого количества признаков [5].

Деревья решений. Модели классификации и регрессии на основе деревьев решений выстраиваются через иерархию условий "если... то", приводящую к решению. В машинном обучении такие условия называются *tests* (тесты). При построении дерева алгоритм перебирает все возможные тесты и находит наиболее информативный с точки зрения прогнозирования значений целевой переменной. Рекурсивное разбиение данных продолжается до тех пор, пока все точки данных на каждом листе дерева решений не будут принадлежать одному значению целевой переменной [12]. В результате такого процесса строится дерево, каждый узел которого соответствует определенному тесту. Лист дерева, который содержит точки, относящиеся к одному и тому же значению целевой переменной, будет называться чистым.

Пусть дано обучающее множество. которое содержит K примеров и N классов. Для оценки эффективности разделения, полученного на основе конкретного атрибута, введем показатель $Q(s \vee t)$, где s – идентификатор разбиения, а t – идентификатор узла. Тогда можно записать [8]:

$$W(s \vee t) = 2 \cdot P_L \cdot P_R \sum_{j=1}^{N} (P(j \vee t_L) - P(j \vee t_R)),$$

где t_L и t_R – левый и правый потомки узла t соответственно, P_L и P_R – отношение количества примеров в левом и правом потомках к их общему количеству в обучающем множестве, $P(j \vee t_L)$ и $P(j \vee t_R)$ – отношение количества примеров класса j в левом и правом потомках к их общему количеству в каждом из них.

Алгоритм дерева решений обладает высокой скоростью работы и не требует масштабирования данных, результаты классификации легко визуализируются [3].

Случайный лес. Основной недостаток деревьев решений – это склонность к переобучению. Одним из способов решения этой проблемы является алгоритм случайного леса [15]. Случайный лес представляет собой набор деревьев решений, в котором каждое дерево немного отличается по параметрам от остальных. Суть заключается в том, что каждое дерево решений хорошо прогнозирует, но, вероятно, переобучается на части данных. Если построить большое количество деревьев решений, которые хорошо отрабатывают и переобучаются с разной степенью, то можно уменьшить переобучение через усреднение их результатов. Для описания указанной стратегии строится множество деревьев решений, которые прогнозируют целевую переменную на должном уровне и отличаются друг от друга. Алгоритм построения случайного леса, состоящего из N деревьев, выглядит следующим образом [6]:

– для каждого $n = 1, \ldots, N$ генерируется выборка X_n, по которой строится решающее дерево:

– по заданному критерию выбирается лучший признак, по которому проводится разбиение дерева (до исчерпания выборки);

– дерево строится до тех пор, пока в каждом из листьев будет не более n_{min} объектов или пока не будет достигнута определенная глубина дерева;

– при каждом разбиении сначала выбирается m случайных признаков из n исходных;

– поиск оптимального разделения выборки проводится только среди них.

Итоговый классификатор выглядит так:

$$a(x) = \frac{1}{N} \sum_{i=1}^{N} b_i(x),$$

иначе говоря, для задачи классификации выбирается решение голосованием по большинству.

Чаще всего алгоритм случайного леса работает лучше, чем одно дерево решений, не требует масштабирования данных, однако плохо обрабатывает разреженные и высокоразмерные данные [3].

Описание разработанного метода

Разработанный метод состоит из следующих этапов:

1. формирование выборки трафика защищаемого приложения;
2. формирование выборки легитимного трафика;
3. разделение выборок на обучающую и тестовую;
4. выделение наиболее информативных признаков (атрибутов сетевого трафика);
5. фаза обучения;
6. оптимизация алгоритма;
7. оценка результатов классификации.

На рис. 1 описан процесс обучения и последующей проверки модели при построении алгоритма классификации.

Формирование выборок сетевого трафика. Для проведения классификации необходимо наличие признаков, характеризующих класс, к которому принадлежит та или иная атака. Исходными данными для решения этой задачи являлись выборки сетевого трафика, собранные с помощью библиотек, обеспечивающих взаимодействие с драйвером сетевого интерфейса.

В ходе выполнения работы было выявлено, что атрибуты сетевого трафика при проведении одного и того же типа атаки могут отличаться в зависимости от программных инструментов, используемых злоумышленником. Вследствие этого в процессе сбора данных для обучения было задействовано несколько программных средств для каждого типа атак (табл. 2), что повысило разнообразие сетевого трафика и позволило точнее выделить информативные признаки.

При построении модели классификации используются метрики из заголовков HTTP-запросов (7 уровень модели OSI), заголовков TCP-пакетов (4 уровень) и IP-пакетов (3 уровень). Это обеспечивает наиболее широкий спектр данных для анализа и, как следствие, более высокую вероятность нахождения признаков, которые могут быть использованы для успешной и эффективной классификации сетевого трафика [13, 14].

Собранный сетевой трафик состоял из нескольких типов данных:

1. трафика различных DDoS-атак, сгенерированных с помощью скриптов и программ, перечисленных в табл. 2;
2. легитимных запросов к защищаемому устройству, в том числе пакетов, необходимых для работы устройства, которыми обмениваются установленные на нем сервисы.

Далее формируются две выборки с данными (обучающая и тестовая) в отношении 70/30. Перед разбиением набор данных был перемешан с помощью генератора псевдослучайных чисел. Это сделано для того, чтобы избежать отбора экземпляров, относящихся только к одному классу (например, если мы возьмем последние 30% экземпляров в качестве тестового набора, то все они будут иметь одинаковую метку, так как сортировка идет именно по ним). Использование тестового набора, содержащего только один класс, не даст возможность объективно судить об обобщающей способности модели классификации. Следует отметить, что был выполнен перебор значений параметров и выбор тех комбинаций, которые дают максимальную точность на тестовом наборе данных, и это не означает, что на новых данных точность будет аналогичной. Если тестовый

Рис. 1. Процесс обучения и проверки модели классификации

набор был применен для настройки параметров, то его использование для оценки качества модели не представляется возможным. По этой же причине данные разбиты на обучающую и тестовую выборки (рис. 2).

В дальнейшем к обучающей выборке применяются алгоритмы классификации, результаты работы которых дают возможность получить необходимые признаки и диапазоны их значений для работы с тестовой выборкой.

Выделение наиболее информативных признаков. Следующей важной процедурой предварительной обработки данных является отбор наиболее информативных признаков. Его цель состояла в отборе наиболее значимых признаков для решения рассматриваемой задачи классификации. Это позволило построить более простые модели классификации с лучшей обобщающей способностью, а следовательно, ускорить работу модели классификации, а также улучшить качество предсказания за счет устранения шумовых признаков [11]. Отбор признаков проводился как вручную, при помощи экспертных знаний (исключались признаки, которые не могут указывать на вредоносный характер трафика), так и при помощи автоматизированного алгоритма.

Автоматизированное вычисление важности каждого признака осуществлялось через отбор признаков на основе модели. Модель машинного обучения с учителем, использованная для отбора признаков, не применялась для создания итоговой модели. Было проведено вычисление определенного показателя важности для всех признаков для последующего ранжирования характеристик по этой метрике.

Отбор признаков на основе модели рассматривал все признаки сразу и поэтому мог обнаружить взаимодействия, когда модель была способна их выявить. По окончании процедуры отбора каждый признак получил значение, которое выражает степень его информативности (сумма всех значений равна 1). Чем выше значение, тем ценнее этот признак для классификации.

В результате получен перечень признаков, которые можно ранжировать в соответствии с их значимостью для решения задачи классификации.

Поиск оптимальных параметров модели. Автоматизированный поиск оптимальных значений параметров и оценка модели проводятся путем использования связки решетчатого поиска с комбинированной проверкой и 5-блочной кросс-валидации [1]. Процесс отбора параметров и последующей оценки модели представлен на рис. 3.

Алгоритм 5-блочной кросс-валидации разбивает данные несколько раз, выстраивая соответствующее количество моделей. Сначала данные разбиваются на 5 частей одинакового размера, называемых блоками. Далее строится последовательность из моделей, где первая модель обучается используя первый блок в качестве тестовой выборки, а остальные блоки (2–5) выполняют роль обучающей выборки (рис. 4). Аналогичные процессы построения моделей повторяются для всех остальных блоков. Первая из пяти частей данных формирует первый блок, вторая из пяти частей данных – второй блок и так далее. Таким образом, вычисляется средняя для всех разбиений величина ошибки на контрольных подвыборках [17,18].

Решетчатый поиск с комбинированной проверкой позволяет осуществить нахождение лучшего набора параметров, доставляющих минимум ошибок перекрестного контроля.

Тестирование. Благодаря описанным выше операциям были построены и оптимизированы классификаторы сетевого трафика на основе трех алгоритмов машинного обучения. Результаты проверки их работы на тестовой выборке представлены в сравнительной таблице (табл. 3). Наилучшие результаты на тестовых выборках показал алгоритм случайного леса (95%)

Таблица 2

Используемые утилиты для генерации DDoS-трафика

Медленные атаки	*torshammer, slowhttptest, sockstress*
Атаки HTTP-flood	*torshammer, goldeneye, ddosim*
Атаки на стек TCP	*hping3, sitekiller, sprut*
Атаки UDP-flood	*hping3, fudp*
Атаки через SSL	*thc-ssl-dos, sslstrip*
Amplification-атаки	*saddam, chargen_amp, ufonet*
Атаки ICMP-flood	*hping3, hyenae*

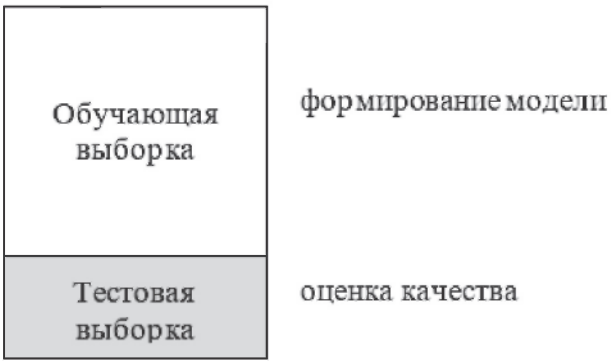

Рис. 2. Разделение на обучающую и тестовую выборки

Рис. 3. Процесс отбора параметров и оценки модели

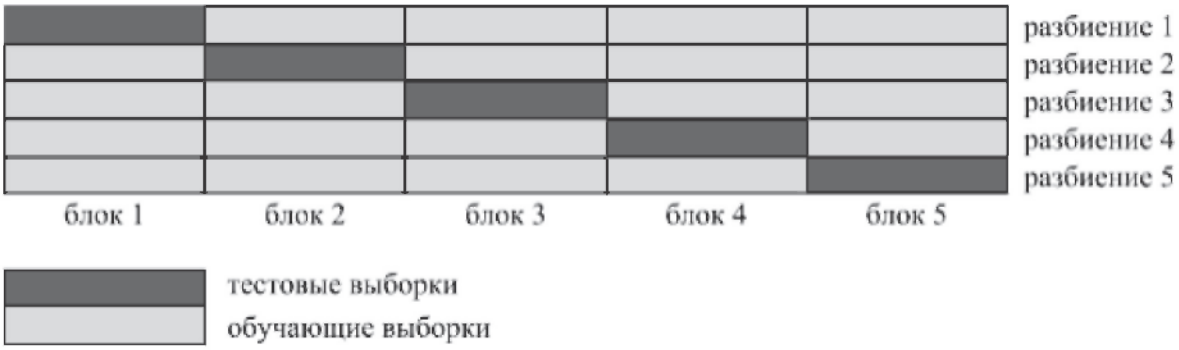

Рис. 4. Разбиение данных при 5-блочной кросс-валидации

Таблица 3

Сравнительная таблица классификаторов сетевого трафика

Алгоритм	До оптимизации	После оптимизации
Дерево решений	86%	91%
k-ближайших соседей	91%	94%
Случайный лес	93%	95%

Во всех трех случаях примененные методы позволили определить наиболее оптимальные параметры моделей и повысить точность классификации сетевого трафика дополнительно на 2–5%.

Заключение

В ходе работы были рассмотрены существующие алгоритмы классификации данных на основе машинного обучения, проведена оценка возможности их применения для детектирования сетевых распределенных атак типа «отказ в обслуживании». Наилучшие результаты на тестовых данных показал алгоритм случайного леса (95%). Результаты исследования применяются в работе по созданию автоматизированной программно-аппаратной платформы защиты от 40 Гбит/с DDoS-атак на основе сетевых плат FPGA PCI-E.

Литература

1. Muller, A. C. Introduction to Machine Learning with Python: A Guide for Data Scientists / Andreas C. Muller, Sarah Guido. – San Francisco: O'Reilly Media, 2016. – P. 268–328

2. Bhattacharyya, D. K. DDoS Attacks: Evolution, Detection, Prevention, Reaction, and Tolerance / Dhruba Kumar Bhattacharyya, Jugal Kumar Kalita. – London: Chapman and Hall/CRC, 2016. – P. 23–70

3. Garreta, R., Moncecchi G. Learning scikit-learn. Machine Learning in Python / R. Garreta. – Birmingham: Packt, 2013. P. 25–60

4. DDoS-атаки и электронная коммерция: современные подходы к защите [Электронный ресурс] / 1-С Битрикс // Сайт Хабрахабр. – Режим доступа: https://habrahabr.ru/company/bitrix/blog/267947, свободный. – Загл. с экрана.

5. Машинное обучение: метод ближайших соседей [Электронный ресурс] / Сайт MachineLearning.ru. – Режим доступа: http://www.machinelearning.ru/wiki/index.php?title=%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_%D0%B1%D0%BB%D0%B8%D0%B6%D0%B0%D0%B9%D1%88%D0%B5%D0%B3%D0%BE_%D1%81%D0%BE%D1%81%D0%B5%D0%B4%D0%B0, свободный. – Загл. с экрана.

6. Обучение на размеченных данных. Случайные леса [Электронный ресурс] / Сайт coursera.orgю – Режим доступа: https://ru.coursera.org/learn/supervised-learning/lecture/bejGu/sluchainyie-liesa, свободный. – Загл. с экранаэ

7. Отчет за 2017 год компании Qrator Labs [Электронный ресурс] / Компания Qrator Labs. – Режим доступа: https://qrator.net/presentations/QratorAnnualRepRus.pdf, свободный. – Загл. с экрана.

8. Технологии анализа данных. Алгоритм CART [Электронный ресурс] / Компания BaseGroup Labs. – Режим доступа: http://docplayer.ru/32842822-Algoritm-cart-m-154.html, свободный. – Загл. с экрана.

9. Отчет по безопасности за 2016 год [Электронный ресурс] / Cisco Systems. – Режим доступа: https://www.cisco.com/c/dam/m/ru_ru/cisco_2016_asr_011116_ru.pdf, свободный. – Загл. с экрана.

10. Воробьева, А. А. История развития программно-аппаратных средств защиты информации / А.А. Воробьева, И.С. Пантюхин.– СПб: Университет ИТМО, 2017. – 62 с.

11. Ерохин, С. Д. Влияние фонового трафика на эффективность классификации приложений методами машинного обучения / С.Д. Ерохин, А.В. Ванюшина // T-Comm: Телекоммуникации и транспорт. – 2017. – Т. 11, № 12. – С. 31–36.

12. Кафтанников, И. Л. Особенности применения деревьев решений в задачах классификации / И.Л. Кафтанников, А.В. Парасич // Вестник ЮУрГУ. Серия «Компьютерные технологии, управление, радиоэлектроника». – 2015. – Т. 15, № 3. – С. 26–32.

13. Зубриенко, Г. А. Методы оптимизации выборки данных для определения аномального трафика / Г.А. Зубриенко, О.Р. Лапонина // International Journal of Open Information Technologies. – 2016. – Vol. 6, No. 3. – P. 1–8.

14. Котов, В. Д. Современное состояние проблемы обнаружения сетевых вторжений / В.Д. Котов, В.И. Васильев // Вестник УГАТУ. – 2012. – Т. 16, № 3. – С. 198–204.

15. Чистяков, С. П. Случайные леса: обзор / С.П. Чистяков // Труды КарНЦ РАН. – 2013. – № 1. – С. 117–136.

16. Мухамедиев, Р. И. Таксономия методов машинного обучения и оценка качества классификации и обучаемости [Электронный ресурс] / Р.И. Мухамедиев, Е.Л. Мухамедиева, Я.И. Кучин // Электронный журнал Cloud of science. – 2015. – Т. 2, №3. – Режим доступа: https://cyberleninka.ru/article/v/taksonomiya-metodov-mashinnogo-obucheniya-i-otsenka-kachestva-klassifikatsii-i-obuchaemosti, свободный. – Загл. с экрана.

17. Воронцов, К. В. Комбинаторный подход к оценке качества обучаемых алгоритмов / К.В. Воронцов // Математические вопросы кибернетики / под ред. О.Б. Лупанов. – М.: Физматлит, 2004. – Т. 13. – С. 5–36.

18. Mullin, M. Complete Cross-Validation for Nearest Neighbor Classifiers / M. Mullin, R. Sukthankar // Proceedings of International Conference on Machine Learning. – 2000. – P. 1137–1145.

19. Zargar, S. T. A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks / Saman Taghavi Zargar, James Joshi and David Tipper // IEEE communications surveys & tutorials 15.4 (2013): 2046-2069.

20. Gupta, Ms. Sh. Application Layer DDOS Attack: A Big Threat / Ms Shaveta Gupta, Dr. Dinesh Grover // Streames Info-Ocean. – June 2016. – Vol. 1, Iss. 1. – P. 30–43.

21. Пантюхин, И. С. Методика проведения постинцидентного внутреннего аудита средств вычислительной техники / И.С. Пантюхин, И.А. Зикратов // Научно-технический вестник информационных технологий, механики и оптики. – 2017. – Т. 17, № 3. – С. 467–474. – doi: 10.17586/2226-1494-2017-17-3-467-474.

22. Pantiukhin, I. Basics of Computer Forensics / I. Pantiukhin, D. Shidakova // Вестник полиции. – 2016. – № 1 (7). – С. 20–29.