МЕТОД ГЛАВНЫХ КОМПОНЕНТ ПРИ ВЫБОРЕ АРХИТЕКТУРЫ НЕЙРОННОЙ СЕТИ АВТОЭНКОДЕРА

DOI: 10.36724/2072-8735-2024-18-10-27-33

Manuscript received 14 September 2024;
Accepted 02 October 2024

Татарникова Татьяна Михайловна,
Санкт-Петербургский государственный университет аэрокосмического приборостроения,
г. Санкт-Петербург, Россия, ***tm-tatarn@yandex.ru***

Ключевые слова: *метод главных компонент, сокращение числа параметров, автоэнкодер, ошибка обучения*

Обсуждается задача применения нейронной сети с архитектурой автоэнкодера для обнаружения сетевых атак. Приводится формализованное описание структуры автоэнкодера. Показано, что одной из трудностей при выборе архитектуры автоэнкодера является подбор сжимающей функции с гарантией восстановления первоначального количества признаков на выходе нейронной сети с допустимой ошибкой. Предложено применять метод главных компонент, как сжимающую функцию с образованием скрытого слоя автоэнкодера. Приведено формальное описание метода главных компонент: математический аппарат вычисления матрицы счетов и матрицы нагрузок, особенности процедур подготовки данных. Предложен итерационный алгоритм поиска главных компонент, позволяющий выбрать число главных компонент с учетом достижения метрик оценивания остаточных значений к минимально допустимому. В качестве метрик статочных значений могут быть величина, указывающая на сумму квадратичных отклонений оригинального вектора входных данных от его проекции в пространство главных компонент или усреднение квадратичных отклонений по всем признакам каждого примера. Применение метода главных компонент позволяет найти матрицу счетов, представляющей собой проекцию признаков, характеризующих образца на пространство главных компонент. Состоятельность применения автоэнкодера для детектирования аномального трафика и идентификации классов атак продемонстрирована на примерах обучения из датасета NSL-KDD, включающем известные типы атак, представляющие четыре группы категорий: отказ в обслуживании, несанкционированное получение прав пользователя, несанкционированное повышение прав пользователя до администратора и зондирование. Получены оценки достоверности классификации сетевых атак, точность классификации, доля истинно положительных решений и F-мера, сочетающая оценки полноты и точности.

Информация об авторе:
Татарникова Татьяна Михайловна, *доктор технических наук, профессор, Санкт-Петербургский государственный университет аэрокосмического приборостроения, институт информационных технологий и программирования, г. Санкт-Петербург, Россия*

Для цитирования:
Татарникова Т.М. Метод главных компонент при выборе архитектуры нейронной сети автоэнкодера // T-Comm: Телекоммуникации и транспорт. 2024. Том 18. №10. С. 27-33.

For citation:
Tatarnikova T.M. (2024). Principal component method in choosing the architecture of the neural network of the autoencoder. *T-Comm*, vol. 18, no.10 pp. 27-33. (*in Russian*)

Введение

Анализ и обработка больших данных стали необходимостью в различных отраслях экономики от медицинских исследований до финансового анализа и промышленной автоматизации. Умение обрабатывать и использовать большие объемы информации становится ключевым конкурентным преимуществом.

С одной стороны, большие данные дают возможность выявить специфические закономерности внутри меньших подгрупп и слабые связи между признаками, а значит делать более детальные прогнозы и решения, принимаемые на их основе. С другой стороны, большие данные требуют длительного процесса предобработки, заключающегося в слиянии разнообразных источников информации и необходимости одновременной оценки и проверки огромного количества параметров, что предсказуемо приводит к увеличению погрешности. Иногда погрешности могут превышать важность настоящих данных, становясь доминирующим. Снятие этой проблемы решается применением методов разреженности к отбору значимых параметров больших данных [1].

Предлагается на этапе предобработки данных применять метод главных компонент (Principal Component Analysis, PCA). Метод позволяет выявить основные корреляции между параметрами многомерных данных, и, соответственно, сократить их размерность, удалив малоинформативные данные. Это помогает повысить производительность моделей машинного обучения, уменьшить переобучение и ускорить вычисления [2].

В работе показана возможность применения метода главных компонент в качестве функции сжатия нейронной сети с архитектурой автоэнкодера и демонстрация работы данного автоэнкодера по распознавание аномального трафика инфокоммуникационных сетей.

Источники аномального трафика

Источниками появления аномалий в сетевом трафике, несущими вредоносный код, являются атаки, вызванные несанкционированными действиями со стороны легитимных пользователей и злоумышленников [3].

Общая классификация сетевых атак включает четыре группы:

– Danial of Service (DoS) – отказ в обслуживании;

– Remote to Local (R2L) – несанкционированное получение прав пользователя;

– User to Root (U2R) – несанкционированное повышение прав пользователя до администратора;

– Probe – Зондирование.

Приведем наиболее характерные признаки атак каждой группы.

Характерной чертой атак, входящих в группу «Отказ в обслуживании» является набор действий, направленных на истощение вычислительных ресурсов атакуемой системы, в следствии чего легитимные пользователи получают отказ в обслуживании. Известными представителями этой группы атак являются: neptune, smurf, land и другие [4].

Neptune – атака, действия которой связаны с отправкой компьютеру-жертве запроса на установление соединения от большого числа других компьютеров без подтверждения установленного соединения. Несмотря на истечение таймаута, позволяющего сбросить служебную запись об установленном соединении, производительность компьютера-жертвы уже начинает стремительно падать и в кое-то время происходит аварийное завершение его работы.

Smurf – атака, при реализации которой от имени компьютера-жертвы рассылаются широковещательные служебные пакеты, требующие ответа. При одновременном ответе компьютеров на требование происходит истощение вычислительных ресурсов компьютера-жертвы и снижение пропускной способности канала.

Land – атака, при которой попытка установления TCP-соединения заканчивается тем, что при передаче пакета с флагом синхронизации SYN компьютер-жертва пытается соединиться сам с собой. Злоумышленник, воспользовавшись тем, что запрос на TCP-соединение передается на открытый порт компьютера имеет возможность установить в соответствующих полях пакета с флагом синхронизации SYN адрес источника равным адресу получателя. Атака приводит к «зависанию» или перезагрузке компьютера. Реализация атаки Land на маршрутизатор способна вывести из работоспособного состояния целый сетевой сегмент.

Суть атак из группы «Попытка несанкционированного получения прав пользователя» заключается в действиях злоумышленника, осуществляющего попытку получить несанкционированный доступ к удаленному компьютеру законного пользователя. Атаки guess_password, multihop являются представителями этой группы.

Guess_password является атакой подбора пароля, которая включает в себя успешное угадывание злоумышленниками пароля из локального или удаленного положения с использованием либо автоматизированного метода, либо иного ручного подхода. Поскольку большинство сетей не настроены таким образом, чтобы требовать сложных паролей, то сетевой доступ может быть потенциально достигнут путем идентификации злоумышленником только одного слабого пароля.

Multihop является атакой взлома почтового сервера, который впоследствии получает доступ к клиенту, где сервер электронной почты находится на том же сервере.

Характерной чертой атак группы «несанкционированное повышение прав пользователя до администратора» является набор действий, направленных на получение информации об уязвимостях компьютерной системы под учетной записью обычного пользователя, чтобы в дальнейшем завладеть правами администратора системы. Известные представители атак этой группы: buffer_overflow, rootkit [5].

Buffer_overflow – атака, при реализации которой осуществляется попытка записать в памяти компьютера-жертвы больше данных, чем она может принять. Поскольку в случае переполнения каждое новое сообщение записывается поверх имеющихся данных, то появляется возможность запустить на исполнение вредоносный код.

Rootkit – атака, реализуемая в виде программы-шпиона с целью предоставления привилегированного доступа к ресурсам компьютера. В настоящее время rootkit чаще всего связывают с вредоносным программным обеспечением, которое, как известно, маскирует свое присутствие и действия от пользователей и системных процессов. Также rootkit не позволяет идентифицировать злоумышленника при получении им основного контроля.

Характерной особенностью атак из группы «Зондирование» является выполнение злоумышленником сканирования интерфейсов компьютера с целью выявления уязвимостей, которые в дальнейшем используются для построения процедуры компрометации. Наиболее известные классы атак этой группы: Ipsweep, Mscan.

Ipsweep – атака построения маршрута в обход элементов защиты. Такая ситуация возможна при использовании алгоритмов маршрутизации от источника в протоколах передачи данных. Злоумышленник, получив доступ к атакованным узлам подменяет адрес в поле пакета «IP-адрес получателя» таким образом, что сообщения отправляются в обход межсетевых экранов.

Mscan – атака сканирования сети на предмет слабых мест или точек входа с целью получения доступа к сетевым ресурсам.

Своевременное выявление аномалии выполняется средствами обнаружения вторжений. Современные системы обнаружения вторжений, функции которых заключаются в детектировании аномального трафика с последующей идентификацией атаки, основаны на поведенческих и/или сигнатурных моделях. Поведенческая модель детектировании аномального трафика, несущего угрозу основана на оценивании отклонения характеристик сетевого узла от его штатной работы [6]. В оценивании характеристик участвуют статистические модели: пороговая, среднего значения, среднеквадратичного отклонения. Если с течением времени наблюдается превышение некоторой характеристики ее порогового значения, то данное обстоятельство может свидетельствовать о наличии атаки. На наличие аномального трафика может указывать, например, возросшая частота запросов и загрузка узла [7]. Сигнатурная модель представляет собой некий паттерн описания атаки и поиска этого паттерна в сетевом трафике. Обе модели предполагают использование нейронных сетей [8, 9].

Нейронная сеть с архитектурой автоэнкодера

Нейронная сеть с архитектурой автоэнкодера представляет собой функцию сжатия – число входных признаков n сокращается до m ($m<n$) с гарантией восстановления первоначального количества признаков n на выходе с допустимой ошибкой ε [10].

На рисунке 1 показана архитектура автоэнкодера.

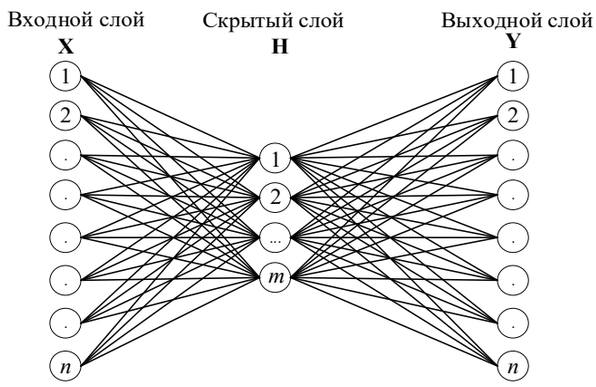


Рис. 1. Архитектура автоэнкодера

Математической модели нейронной сети с архитектурой автоэнкодера соответствует следующая запись:

$$\mathbf{X} \in \mathbf{R}^n = \mathbf{XH} \in \mathbf{R}^m = \mathbf{Y},\ n<m. \quad (1)$$

где вектор **X** задет входные признаки; вектор **Y** соответствует выходным признакам; вектор **H** соответствует сжатому пространству.

И функциями (2-4):

$$f:\ \mathbf{X}\rightarrow\mathbf{H};\ g:\mathbf{H}\rightarrow\mathbf{Y}, \quad (2)$$

$$f, g = \arg\ \min \left\| X - \left[f(g(X)) \right] \right\|^2. \quad (3)$$

$$\mathbf{H} = \sigma(\mathbf{WX} + \mathbf{B}), \quad (4)$$

где **σ** задает функцию активации нейронов; матрица **W** хранить весовые коэффициенты нейронов; вектор **B** устанавливает смещение.

Процедура восстановления представляется аналогично сжатию:

$$\mathbf{Y} = \sigma'(\mathbf{W}'\mathbf{h} + \mathbf{B}'), \quad (5)$$

при этом нет математической зависимости, устанавливающей связь параметров σ', $\mathbf{W}'$, $\mathbf{B}'$ с соответствующим параметрами **σ**, **W**, **B**.

Ошибку восстановления после процедуры сжатия запишем в виде

$$\varepsilon(\mathbf{X},\mathbf{Y}) = \left\| \mathbf{X} - \mathbf{Y} \right\|^2 = \left\| \mathbf{X} - \sigma'(\mathbf{W}'(\sigma(\mathbf{WX} + \mathbf{B})) + \mathbf{B}') \right\|^2. \quad (6)$$

Одним из основных способов уменьшения размерности данных, сохраняя при этом наибольшую информацию является метод главных компонент (Principal component analysis, PCA).

Формализация метода главных компонент

Применение метода главных компонент позволяет с приемлемой точностью воспроизвести исходные данные, при этом не требуются специфические предположения относительно анализируемых переменных.

Пусть матрица X хранит примеры обучения, где число строк I задает количество примеров обучения и число столбцов J – количество признаков.

Введем формальные переменные t_a, a=1, …, A, через которые запишем линейные уравнения для имеющихся примеров обучения x_j, j=1, …, J: $t_a = p_{a1}x_1 + \ldots + p_{aj}x_j$

Применение метода PCA позволяет представить матрицу входных признаков **X** как произведение матриц **T**, размерностью $I \times A$ и **P** размерностью $J \times A$, где $A<I$ и $A<J$, и матрицы остатков **E**:

$$\mathbf{X} = \mathbf{TP}^t + \mathbf{E} = \sum_{a=1}^{A} t_a p_a^t + \mathbf{E}, \quad (7)$$

где A – число главных компонент; t_a – число столбцов матрицы **T**; p_a – число столбцов матрицы **P**.

При добавлении новых признаков и новых примеров обучения не требуется пересчета матриц **T** и **P** благодаря свойству ортогональности главных компонент: при добавлении новых признаков просто добавляются новые столбцы и при

добавлении новых образцов (примеров обучения) – новые строки.

Подготовка данных

Применению метода PCA предшествует этап подготовки данных, который заключается в шкалировании, кодировании и нормировании.

Шкалирование подразумевает приведение значений каждого признака к единой шкале: номинальной, порядковой, интервальной и т.д.

Кодирование заключается в переводе символьных данных в числовой формат, как правило, к порядковой шкале: соответственно в бинарной классификации код одного класса – 0, код другого класса – 1; в множественной классификации количество числовых полей задается равным количеству классов.

Нормирование устраняет дисбаланс значений признаков, образовавшийся после кодирования таким образом, что каждый признак стремится к нормальному распределению.

В результате выполнения этапа подготовки данных все значения признаков принадлежат одному диапазону их изменения, например [0; 1], что позволяет свести их вместе в одну модель и обеспечивает корректную работу вычислительных алгоритмов.

$$\tilde{x}_{ij} = \frac{\left(x_{ij} - m_j\right)}{s_j}. \qquad (8)$$

где x_{ij} – значение j-го признака у i-го примера (образца); m_j – среднее арифметическое j-го признака; s_j – стандартное отклонение j-го признака.

Алгоритм поиска главных компонент

На каждой итерации метода PCA рассчитывается одна главная компонента. Суть алгоритма сводится к построению матрицы счетов **T** и матрицы нагрузок **P**.

На этапе предварительной подготовки данных матрица **X** после шкалирования, кодирования и нормирования превращается в матрицу остатков $\mathbf{E}_0$. Число главных компонент устанавливается равным нулю (a=0).

Дальнейшие шаги алгоритма:

1. Выбор начальных значений вектора **t**.
2. Определение вектора $\mathbf{p}^t = \frac{\mathbf{t}^t\mathbf{E}_a}{\mathbf{t}^t\mathbf{t}}$.
3. Нормирование вектора $\mathbf{p} = \frac{\mathbf{p}}{\sqrt{\mathbf{p}^t\mathbf{p}}}$.
4. Обновление вектора $\mathbf{t} = \frac{\mathbf{p}\mathbf{E}_a}{\mathbf{p}^t\mathbf{p}}$.
5. Проверка условия сходимости оригинального вектора $\mathbf{x}_i$ и его проекции $\mathbf{x}_i'$. Если сходимость не достигнута, то число главных компонент a принять равным (a+1), найти матрицу остатков $\mathbf{E}_{a+1}=\mathbf{E}_a-\mathbf{T}\mathbf{P}^t$ и выполнить переход к шагу 2 алгоритма. Если сходимость достигнута, то выполнить вывод результата: **t**, t_a, **p**, p_a.

Далее строится новая матрица счетов $\mathbf{T}_{new} = \mathbf{X}_{new}\mathbf{P}$, в которой $\mathbf{X}_{new}$ содержит новые образцы как результаты проекции образцов **X** на пространство главных компонент размерности A. Матрица нагрузок **P** также является новым преобразованием перехода от базиса исходного J-мерного пространства образцов $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_J$ к базису пространства главных компонент $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_A$.

Разложение матрицы **X** методом PCA представляет собой итерационную процедуру, которую можно прекратить при достижении нужного числа a=A компонент. В результате, получаем матрицу $\hat{\mathbf{X}} = \mathbf{T}\mathbf{P}^t$, которая отличается от матрицы **X.** Эта разница определяется как матрица остаточных значений $\mathbf{E} = \mathbf{X} - \hat{\mathbf{X}}$.

Метриками оценивания остаточных значений могут быть:

– величина ν_i – сумма квадратичных отклонений оригинального вектора $\mathbf{x}_i$ от его проекции $\mathbf{x}_i'$ в пространстве главных компонент:

$$\nu_i = \sum_{j=1}^{J} e_{ij}^2. \qquad (9)$$

– величина d_i – усреднение ν_i по всем признакам i-го примера:

$$d_i = \frac{1}{J}\sum_{j=1}^{J} e_{ij}^2. \qquad (10)$$

При $\nu_i \to$min или $d_i \to$ min очевидно приближение оригинального вектора $\mathbf{x}_i$ к его проекции $\mathbf{x}_i'$.

Результаты эксперимента по обнаружению аномального трафика автоэнкодером

Обучение автоэнкодера со срытым слоем, полученным методом PCA выполнялось на открытом наборе данных – датасете NSL-KDD 2009 [11]. Датасет является предобработанным, то есть в нем отсутствуют избыточные примеры обучения и дубликаты. Тем не менее некоторые классы несбалансированные, что требует выработки механизма обратного возвращения к уже задействованным в обучении примерам или отказа от них.

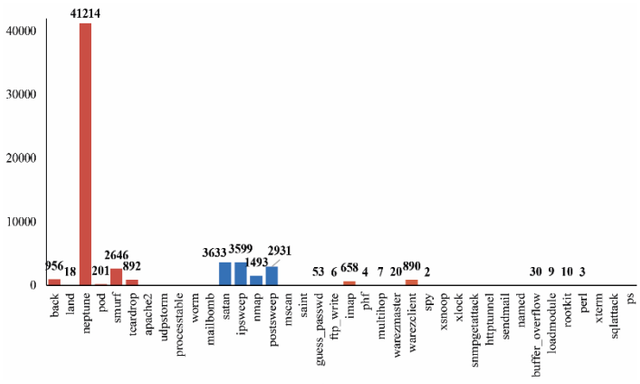
Типы атак сгруппированы в категории Probe, DoS, U2R и R2L [12]:

- Примеры атак группы DoS: Smurf, Land, Neptune, Apache2, Back, Pod, Processtable, Teardrop, Udpstorm, Worm;
- Примеры атак группы R2L: Guess_Password, Multihop, Ftp_write, Httptunnel, Mailbomb, Named, Phf, Sendmail, Snmpguess, Snmpgetattack, Warezmaster, Xlock, Xsnoop;
- Примеры атак группы U2R: Buffer_overflow, Rootkit, Loadmodule, Perl, Ps, Sqlattack, Xterm;
- Примеры атак группы Probe: Ipsweep, Mscan, Nmap, Portsweep, Satan, Saint.

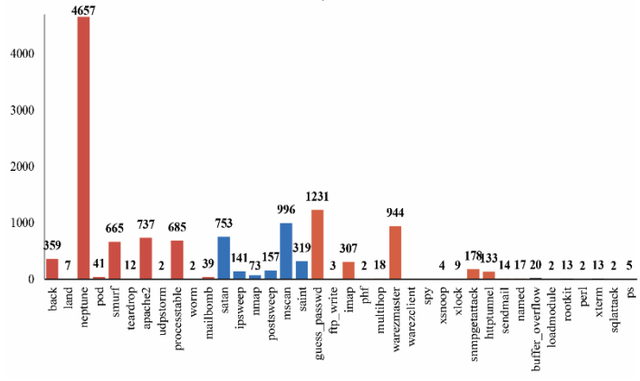
Датасет разбит на обучающий набор данных, который содержит 21 пример обучения и тестовый набор данных, который содержит 37 примеров обучения. На рисунке 2 приведены распределения сетевых атак по группам.

Сетевой трафик, не несущий вредоносный код (атаку) представлен в обучающем наборе 67 343 примерами. Вместе с образцами аномального трафика – классами атак, объем датасета составил 126 620 примера обучения.

В тестовом наборе данных получилось 9 711 примера записей сетевого трафик, не несущего вредоносный код, что вместе с образцами аномального трафика составило 22 850 примера.



а)



б)

Рис. 2. Распределение сетевых атак по группам:
а – в обучающем наборе данных; *б* – в тестовом наборе данных

В целях ликвидации несбалансированности классов были удалены примеры обучения малых объемов. Таким образом осталось 17 классов – видов атак (табл. 1).

Таблица 1

Классы атак из набора данных NSL-KDD

Название класса	Метка класса	Количество примеров
Smurf	0	164091
Normal	1	60593
Neptune	2	58001
Snmpgetattack	3	7741
Mailbomb	4	5000
Guess_passwd	5	4367
Snmpguess	6	2406
Satan	7	1633
Warezmaster	8	1602
Back	9	1098
Mscan	10	1053
Apache2	11	794
Processtable	12	759
Saint	13	736
Portsweep	14	354
Ipsweep	15	306
Httptunnel	16	158

Обучение модели реализовано с помощью библиотек tensorflow и keros языка python в оболочке KerasClassifier [13].

В таблице 2 приведены значения accuracy – достоверности классификации атак.

Таблица 2

Оценка достоверности классификации сетевых атак

Метка класса	Название класса	Accuracy
0	Smurf	99.5
1	Normal	92.0
2	Neptune	99.9
3	Snmpgetattack	99.9
4	Mailbomb	94.9
5	Guess_passwd	97.0
6	Snmpguess	97.9
7	Satan	81.8
8	Warezmaster	98.1
9	Back	97.7
10	Mscan	98.1
11	Apache2	99.7
12	Processtable	98.4
13	Saint	**100,0**
14	Portsweep	61.7
15	Ipsweep	79.3
16	Httptunnel	**100,0**

На рисунке 3 приведены характеристики эффективности классификатора, основанного на автоэнкодере: точность (Precision), доля истинно положительных решений (Recall) и F-мера, сочетающая оценки полноты и точности (Fscore). Характеристики эффективности классификатора и способы их оценивания приведены в [14, 15].

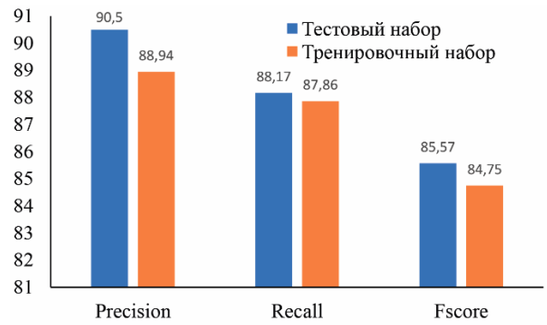


Рис. 3. Результаты эффективности автоэнкодера по обнаружению сетевых атак

Эксперимент показал, что для обнаружения аномалии сетевого трафика обученному автоэнкодеру требуется 0,17 с, что меньше средней продолжительности любой из рассмотренных атак.

Заключение

Обнаружение аномального трафика в инфокоммуникационных сетях является актуальной задачей. Источником аномального трафика являются атаки, вызванные несанкционированными действиями со стороны легитимных пользователей и злоумышленников.

Обнаружение аномального трафика решается средствами систем обнаружения вторжений, среди вариантов реализации которых приобретают популярность технологии искусственного интеллекта и нейронных сетей в частности.

С увеличением числа признаков обучение нейронных сетей может занимать много времени, что при обнаружении сетевых атак в реальном времени недопустимо. К тому же на практике некоторые из архитектур нейронных сетей требуют значительных объемов вычислительных ресурсов для обучения.

В работе предлагается использовать нейронной сети с архитектурой автоэнкодера для обнаружения сетевых атак. Скрытый слой автоэнкодера определяется с помощью метода главных компонент, суть которого заключается в сокращении количества исходных признаков до числа главных компонент. Рассмотрена последовательность шагов алгоритма, вычисления главных компонент.

Эффективность применения автоэнкодера для решения задачи классификации аномального трафика демонстрируется результатами обучения на выборках исходных данных, содержащем в своем составе 17 классов после исключения несбалансированных классов.

Достоверность классификации сетевых атак оценивалась такими метриками как достоверность классификации, точность классификации, доля истинно положительных решений и F-мера. Достоверность классификации выше 90% показана для атак, число примеров обучения которых составляет более 400, исключение коснулось только атаки типа httptunnel. Объяснение таким результатам можно дать только принадлежностью атак к одной категории, а значит схожестью признаков. Точность классификатора, основанного на автоэнкодере составила 90%, доля истинно положительных решений – 88% и F-мера – 85% на обучаемом наборе данных, и на 1-2% ниже значений этих же показателей эффективности классификатора на тестовом наборе.

Модель автоэнкодера со скрытым слоем, найденным методом главных компонент позволяет обнаружить атаку за 0,17 с, что меньше времени продолжительности реализации любой из рассматриваемых в работе атак.

Полученные результаты показывают, что обученный автоэнкодер можно применять в качестве основы построения системы обнаружения вторжений. Такая система обнаружения вторжений позволит не только детектировать аномальный трафик, но и идентифицировать атаку, распознавать которую обучен автоэнкодер.

Литература

1. *Самойлова И.А.* Технологии обработки больших данных // Молодой ученый. 2017. № 49 (183). С. 26-28.

2. *Кабаков Р.И.* R в действии. Анализ и визуализация данных в программе R. М.: ДМК Пресс, 2016. 588 с.

3. *Бачманов Д.А., Очередько А.Р., Путято М.М., Макарян А.С.* Исследование вопросов совершенствования систем защиты от ddos-атак на основе комплексного анализа современных механизмов противодействия // Прикаспийский журнал: управление и высокие технологии. 2021. №1 (53). [Электронный ресурс]. Режим доступа: https://hi-tech.asu.edu.ru/?articleId=1294 (дата обращения: 24.09.2024) DOI: 10.21672/2074-1707.2021.53.1.063-074

4. *Михайлова У.В., Афанасьева М.В.* Обнаружение DOS или DDOS атак // 77-я международная научно-техническая конференция "Актуальные проблемы современной науки, техники и образования". 2019. Т. 1. С .426-426.

5. *Suroto S.* A review of defense against slow HTTP attack // JOIV: International Journal on Informatics Visualization, 2017, no. 1(4), pp. 127-134.

6. *Wu S.X., Banzhaf W*. The Use of Computational Intelligence in Intrusion Detection Systems: A Review // Applied Soft Computing. 2010. Vol. 10(1), pp. 1-35.

7. *Созыкин А.В*. Обзор методов обучения глубоких нейронных сетей // Вестник ЮУрГУ, Серия: Вычислительная математика и информатика. 2017. Т.6. № 3. С. 28-59.

8. *Tatarnikova T.M*. Restricting data leakage through non-obviouse features of Android 5 smartphone // Informationno-Upravliaiushchie Sistemy. 2019. Vol. 5, pp. 25-29.

9. Сафронова Е.О., Жук Г.А. Применение искусственных нейронных сетей для прогнозирования DoS атак // Молодой ученый. 2019. № 23. С. 27-30.

10. *Шелухин О.И., Филинова А.С*. Сравнительный анализ алгоритмов обнаружения аномалий трафика методами дискретного вейвлет-анализа // T-Comm. 2014. Т. 9. С. 89-97.

11. A Deeper Dive into the NSL-KDD Data Set – Towards Data Science [Электронный ресурс]. Режим доступа: https://towardsdatascience.com/a-deeper-dive-into-the-nsl-kdd-data-set-15c753364657 (дата обращения: 10.09.2024).

12. *Ingre B., Yadav A., Soni A.K.* Decision Tree Based Intrusion Detection System for NSL-KDD Dataset // Proceedings of the International Conference on Information and Communication Technology for Intelligent Systems (ICTIS, Ahmedabad, India, 25-26 March 2017). Cham: Springer. 2017. Vol. 2, pp. 207-218.

13. *Ertam F., Aydin G*. Data Classification with Deep Learning Using Tensorflow // International Conference on Computer Science and Engineering. 2017, pp. 755-758.

14. *Татарникова Т.М. Бимбетов Ф., Богданов П.Ю.* Выявление аномалий сетевого трафика методом глубокого обучения // Известия СПбГЭТУ ЛЭТИ. 2021. № 4. С. 36-41.

15. *Татарникова Т.М., Богданов П.Ю.* Метрические характеристики обнаружения аномального трафика в сетях интернета вещей // T-Comm: Телекоммуникации и транспорт. 2022. Т. 15. №1. С. 15-21.

PRINCIPAL COMPONENT METHOD IN CHOOSING THE ARCHITECTURE OF THE NEURAL NETWORK OF THE AUTOENCODER

Tatyana M. Tatarnikova, *St. Petersburg State University of Aerospace Instrumentation, Saint-Petersburg, Russia,* ***tm-tatarn@yandex.ru***

Abstract

Detection of abnormal traffic in infocommunication networks is an urgent task. The source of abnormal traffic are attacks caused by unauthorized actions on the part of legitimate users and intruders. Detection of abnormal traffic is solved by means of intrusion detection systems, among the implementation options of which artificial intelligence technologies and neural networks in particular are gaining popularity. With an increase in the number of features, training neural networks can take a lot of time, which is unacceptable when detecting network attacks in real time. In addition, in practice, some of the neural network architectures require significant amounts of computing resources for training. The paper proposes to use a neural network with an autoencoder architecture to detect network attacks. The hidden layer of the autoencoder is determined using the principal component method, the essence of which is to reduce the number of original features to the number of principal components. The sequence of steps of the algorithm, calculating the principal components, is considered. The efficiency of using an autoencoder to solve the problem of classifying abnormal traffic is demonstrated by the results of training on samples of initial data containing 17 classes after excluding unbalanced classes. The reliability of network attack classification was assessed by such metrics as classification reliability, classification accuracy, proportion of true positive decisions, and F-measure. Classification reliability above 90% was shown for attacks with more than 400 training examples, the only exception was the httptunnel attack. Such results can only be explained by the fact that the attacks belong to the same category, and therefore by the similarity of features. The accuracy of the classifier based on the autoencoder was 90%, the proportion of true positive decisions was 88%, and the F-measure was 85% on the training data set, and 1-2% lower than the values of the same classifier efficiency indicators on the test set. The autoencoder model with a hidden layer found by the principal component method allows detecting an attack in 0.17 s, which is less than the duration of the implementation of any of the attacks considered in the work. The obtained results show that the trained autoencoder can be used as a basis for building an intrusion detection system. Such an intrusion detection system will allow not only to detect abnormal traffic, but also to identify the attack that the autoencoder is trained to recognize.

Keywords: *principal component method, reduction of the number of parameters, autoencoder, training error.*

References

[1] I.A. Samoilova, "Big data processing technologies," Young scientist, 2017, no. 49 (183), pp. 26-28. (in Russian)
[2] R.I. Kabakov, "Data analysis and visualization in the R program.," Moscow: DMK Press, 2016, 588 p. (In Russian)
[3] D.A. Bachmanov, A.R. Ocheredko, M.M. Putyato, A.S. Makaryan, "Research of the issues of improvement of protection systems against ddos-attacks based on the comprehensive analysis of modern interaction mechanisms," *Caspian Journal: Control and High Technologies*, 2021, no. 1 (53). Available at: https://hi-tech.asu.edu.ru/?articleId=1294 (accessed: 24 September 2024)
[4] U.V. Mikhailova, M.V. Afanasyeva, "Detection of DOS or DDOS attacks," *77th International Scientific and Technical Conference "Actual Problems of Modern Science, Technology and Education*", 2019, vol. 1, pp. 426-426.
[5] S. Suroto, "A review of defense against slow HTTP attack," *JOIV: International Journal on Informatics Visualization*, 2017, vol. 1(4), pp. 127-134.
[6] S.X. Wu, W. Banzhaf, "The Use of Computational Intelligence in Intrusion Detection Systems: A Review," *Applied Soft Computing*, 2010, vol. 10(1), pp. 1-35.
[7] A.V. Sozykin, "An Overview of Methods for Deep Learning in Neural Networks," *Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering*, 2017, vol. 6, no. 3, pp. 28-59. (in Russian) DOI: 10.14529/cmse170303.
[8] T.M. Tatarnikova, "Restricting data leakage through non-obviouse features of Android 5 smartphone," *Informationno-Upravliaiushchie Sistemy*, 2019, vol. 5, pp. 25-29.
[9] E.O. Safronova, G.A. Zhuk, "Application of artificial neural network for predicting DoS attack," *Young Scientist*, 2019, no. 23, pp. 27-30.
[10. O.I. Sheluhin, A.S. Filinova, "The comparative analysis of detection algorithms detection of traffic anomalies methods of the discrete wavelet-analysis," *T-Comm*, 2014, vol. 9, pp. 89-97.
[11] A Deeper Dive into the NSL-KDD Data Set – Towards Data Science. Available at: https://towardsdatascience.com/a-deeper-dive-into-the-nsl-kdd-data-set-15c753364657 (accessed: 10 September 2024).
[12] B. Ingre, A. Yadav, A.K. Soni, "Decision Tree Based Intrusion Detection System for NSL-KDD Dataset," *Proceedings of the International Conference on Information and Communication Technology for Intelligent Systems* (ICTIS, Ahmedabad, India, 25-26 March 2017). Cham: Springer, vol. 2, pp. 207-218.
[13] E. Fatihand, G. Aydin, "Data Classification with Deep Learning Using Tensorflow," *International Conference on Computer Science and Engineering*, 2017, pp. 755-758.
[14] T.M. Tatarnikova, F. Bimbetov, P.Yu. Bogdanov, "Detection of network traffic anomalies by deep learning," Izvestiya SPbGETU LETI, 2021, no. 4, pp. 36-41 (In Russian)
[15] T.M. Tatarnikova, P.Yu. Bogdanov, "Metric characteristics of anomalous traffic detection in internet of things," *T-Comm*, 2022, vol. 15, no.1, pp. 15-21 (in Russian)

Information about author:

Tatyana M. Tatarnikova, *Doctor of Technical Sciences, Professor, St. Petersburg State University of Aerospace Instrumentation, Institute of Information Technology and Programming, St. Petersburg, Russia*