УДК 004.9
DOI: 10.25559/SITITO.18.202201.144-151

Научная статья

Обнаружение вредоносной активности в зашифрованном трафике, представленном в виде временных рядов

М. С. Полянская
ФГБОУ ВО «Московский государственный университет имени М. В. Ломоносова», г. Москва, Российская Федерация
Адрес: 119991, Российская Федерация, г. Москва, ГСП-1, Ленинские горы, д. 1
m.s.polyanskaya@mail.ru

Аннотация
В данный момент большая часть трафика в Интернете зашифрована; вредоносные программы также всё чаще используют шифрование. Чтобы анализировать зашифрованный трафик на предмет вредоносной активности, используются его метаданные. За «единицу» трафика принимается поток – соединение между двумя хостами. Тема данной работы – анализ зашифрованного трафика, представленного в виде временных рядов, с помощью машинного обучения. Этот подход рассматривается в сравнении с более традиционным подходом к классификации потоков. Задача рассмотрена в контексте обучения с учителем и без учителя. Также поставлена задача принятия решения о наличии заражения на хосте по совокупности данных, и описана модель детектора заражения. Эксперименты проводились на примере сетевой активности вируса-шифровальщика. Для анализа временных рядов применялись специализированные инструменты: рекуррентные и конволюционные нейросети, алгоритм динамической трансформации временной шкалы.

Ключевые слова: системы обнаружения вторжений, временные ряды, машинное обучение

Автор заявляет об отсутствии конфликта интересов.

Для цитирования: Полянская М. С. Обнаружение вредоносной активности в зашифрованном трафике, представленном в виде временных рядов // Современные информационные технологии и ИТ-образование. 2022. Т. 18, № 1. С. 144-151. doi: https://doi.org/10.25559/SITITO.18.202201.144-151

Original article

Detection of Malicious Activity in Encrypted Traffic Presented as a Time Series

M. S. Polyanskaya
Lomonosov Moscow State University, Moscow, Russian Federation
Address: 1 Leninskie gory, Moscow 119991, GSP-1, Russian Federation
m.s.polyanskaya@mail.ru

Abstract

At the moment, traffic on the Internet is mostly encrypted; malware is also increasingly using encryption. To scan encrypted traffic for malicious activity, its metadata is used. For that purpose, traffic is divided into flows – sessions between two hosts. This paper is devoted to machine learning for analysis of encrypted traffic presented in the form of time series. This approach is considered in comparison with a more traditional approach to flow classification. The task is considered in the context of both supervised and unsupervised machine learning. Regarding decision-making on whether the host is infected as a whole, a model of a malware detector is proposed. The experiments were conducted on the case study of the network activity of ransomware. Specialized tools were used to analyze time series: recurrent and convolutional neural networks, dynamic time warping.

Keywords: intrusion detection systems, time series, machine learning

The author declares no conflict of interest.

For citation: Polyanskaya M.S. Detection of Malicious Activity in Encrypted Traffic Presented as a Time Series. *Sovremennye informacionnye tehnologii i IT-obrazovanie = Modern Information Technologies and IT-Education.* 2022; 18(1):144-151. doi: https://doi.org/10.25559/SITITO.18.202201.144-151

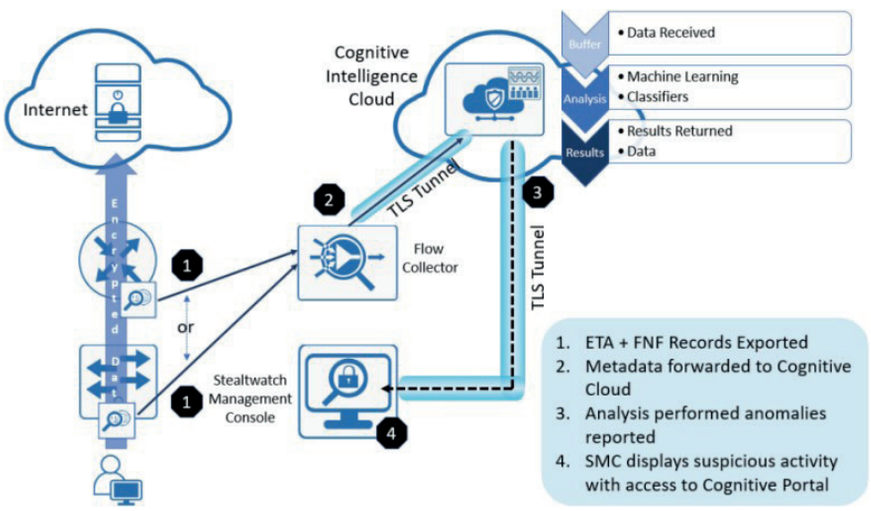
Введение

Согласно данным Google Transparency Report, доля зашифрованного трафика в Интернете постоянно возрастает: от приблизительно 50% в 2014 году до 95% по состоянию на январь 2022 года. 96% наиболее посещаемых сайтов по умолчанию используют протокол HTTPS (TLS). Ещё стремительнее растёт использование шифрования вредоносными программами: за 2020 год на 500% выросло число вирусов-вымогателей, использующих зашифрованный трафик.

Из зашифрованного трафика можно извлечь различные статистические метаданные и некоторые незашифрованные данные, прежде всего – данные инициализации: TLS-рукопожатия, KEX для SSH, IKEv2 для IPSec.

Большинство коммерческих систем сетевой защиты – Intrusion Detection Systems (IDS) – являются сетевыми (NIDS). В отличие от хостовых IDS (HIDS), которые наблюдают за состоянием и событиями внутри хоста, NIDS просматривают трафик от нескольких хостов, которые присоединены к сетевому сегменту. У современных IDS бывает сложная архитектура с удалёнными вычислениями (Рис. 1), и все каналы требуют надёжного шифрования.

В свете вышесказанного, перспективна задача обнаружения вредоносной активности в зашифрованном трафике.



Р и с. 1. Архитектура продукта Cisco ETA[1]: все вычисления производятся в облаке; «локальная» IDS состоит из сборщика потоков (flow collector) и консоли управления (StealthWatch Management Console)

F i g. 1. CiscoETA product architecture: all computing is done in the cloud; The "local" IDS consists of a flow collector and a management console

В данной работе предлагается обрабатывать трафик как совокупность временных рядов. В главе 2 представлен обзор методов машинного обучения, специфичных для временных рядов. В главе 3 – обзор признаков для описания трафика, выделенных исследователями. В главе 4 раскрывается авторский подход и обсуждаются результаты на примере активности вируса-шифровальщика. Глава 5 посвящена моделированию классификатора трафика и принятию решения о заражении по совокупности данных.

Обзор методов машинного обучения

Классификация временных рядов

Временной ряд – это упорядоченная последовательность значений каких-либо показателей за несколько периодов времени.

Методы машинного обучения для задачи классификации временных рядов [1]:

Традиционные:

1. Метрические (distance-based) методы.
   Пример: Dynamic Time Warping (DTW). Основная идея: введение специальной метрики для временных рядов, нечувствительной к временным сдвигам. По этой метрике применяется метод k ближайших соседей. Также хорошо подходит для кластеризации и поиска выбросов.
2. Метод, основанный на шейплетах (shapelets) – шаблонах (patterns), наличие которых во временном ряду позволяет судить о принадлежности ряда к классу. По ряду проходит скользящее окно размером с шейплет, вычисляется евклидово расстояние до каждого шейплета; расстояние от ряда до шейплета – минимальное из этих расстояний. Набор шейплетов выбирается так, чтобы наилучшим образом разделять обучающую выборку.
3. Статистические (feature-based) методы.
   Пример: Bag-of-SFA-Symbols (BOSS), Bag-of-Patterns (BOP) – методы, вдохновлённые подходом «мешок слов» (bag

[1] Encrypted Traffic Analytics: Solutions Adoption Prescriptive Reference – Design Guide. Cisco Systems, 2019. [Электронный ресурс]. URL: https://www.cisco.com/c/dam/en/us/td/docs/solutions/CVD/Campus/eta-design-guide-2019oct.pdf (дата обращения 14.12.2021).

of words) для обработки естественных языков. Основная идея: после преобразований исходные данные разбиваются на равновероятные отрезки (бины), которые переводятся в символьные метки и учитываются как слова в «мешке слов». Полученные вектора классифицируются традиционными методами.

Методы детально разобраны в статье [2].

Глубокое обучение:

Рассмотрим нейросети, которые специально предназначены для обработки последовательных данных.

1. Simple RNN – простейший пример рекуррентной нейросети (Recurrent Neural Network, RNN). RNN анализируют временные ряды, вообще говоря, переменной длины t. Они обладают скрытым состоянием $h(t)$ – памятью (Рис. 2). В случае Simple RNN,

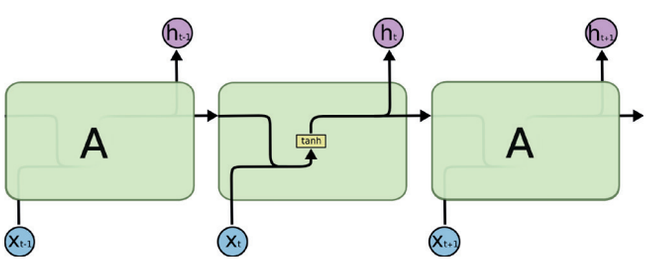
$h(t) = tanh(W_{hh}x(t) + W_{hx}h(t-1) + b_h)$, $y(t) = h(t)$,

где $x(t)$ – входная последовательность, W_{hh}, W_{hx}, b_h – настраиваемые параметры,

$y(t)$ – выходная последовательность.

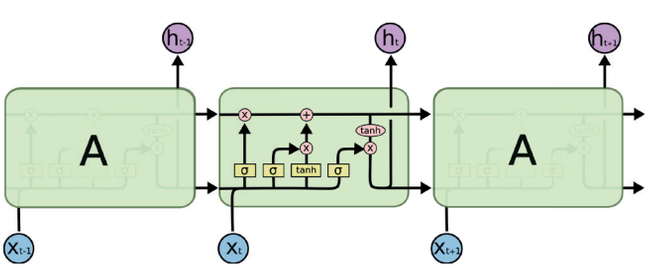
2. Long Short-Term Memory (LSTM) – усовершенствованная версия RNN с долгосрочной памятью (Рис. 3).

Помимо $h(t)$, у её ячейки A есть второй выход – состояние ячейки (cell state) $C(t)$. Нейроны подразделяются на «ворота» (gates), пропускающие или не пропускающие информацию: forget gate, input gate, output gate.



Р и с. 2. Архитектура Simple RNN[2]

F i g. 2. Simple RNN architecture



Р и с. 3. Архитектура LSTM[3]

F i g. 3. LSTM architecture

3. Convolutional Neural Network (CNN) – архитектура, обычно используемая для анализа изображений. Принимает на вход тензор. Состоит из слоёв свёртки (convolution) и подвыборки (pooling). Свёртка выделяет основные черты входа, а подвыборка уменьшает размерность.

Временной ряд следует представить как тензор , где – количество отсчётов времени, а – количество признаков.

CNN более эффективны вычислительно, но хорошо обнаруживают лишь локальные паттерны. CNN по определению принимает на вход тензор, то есть хуже интерпретирует ряды разной длины (которые могут быть обработаны ей после дополнения). Могут использоваться в комбинации с RNN, чтобы предварительно выделить паттерны и уменьшить размерность.

Задача обнаружения выбросов во временных рядах

Помимо классификации (обучения с учителем), для обнаружения вредоносного трафика актуальна задача обнаружения выбросов (без учителя), так как трафик часто бывает не размечен.

Наиболее популярны для этой задачи метрические методы: выбросом называется точка, не попавшая в кластеры – плотные области. Применим метод кластеризации DBSCAN с вышеупомянутой метрикой DTW.

Расстояние DTW между двумя временными рядами длин вычисляется по формуле:

$$DTW(x,y) = \min_{\pi} \sqrt{\sum_{(i,j)\in\pi} d\left(x_i, y_j\right)^2}$$

где $\pi_k = (i_k, j_k)$ – пары моментов времени $(0 \le i_k < n, 0 \le j_k < m)$

$\pi_0 = (0,0), \pi_K = (n-1, m-1)$,

для всех k: $i_{k-1} \le i_k \le i_{k-1} + 1, j_{k-1} \le j_k \le j_{k-1} + 1$,

d – это введённое расстояние между точками (например, расстояние Говера для смешанных – числовых и категориальных – данных).

Признаки для описания трафика

Для обнаружения вредоносной активности наиболее информативна полезная нагрузка. Её в общем случае анализируют на уровне байтов (символов): рассчитывается распределение байтов, либо применяются подходы из обработки естественного языка (natural language processing, NLP), где пакет – это «текст» из байтов-«слов». Также могут анализироваться заголовки, типовые события (сообщения об ошибках), сигнатуры атак.

Существуют криптографические методы, позволяющие проанализировать полезную нагрузку без раскрытия: полностью гомоморфное шифрование и многосторонние конфиденциальные вычисления, но они очень неэффективны и требуют модификации (дискретизации) нейросетей с некоторой потерей точности [3-5], [9-20].

Остановимся на анализе метаданных. Метаданные – это всевозможные незашифрованные свойства трафика. Их можно разделить на группы:

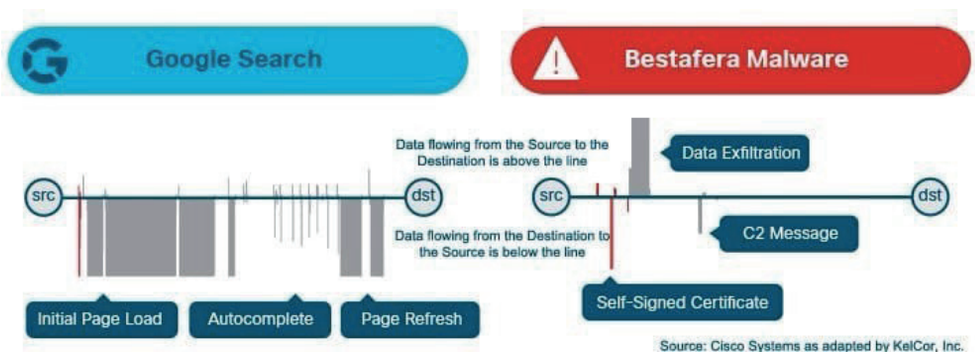
[2] Olah C. Understanding LSTM Networks. GitHub, 2015. [Электронный ресурс]. URL: https://colah.github.io/posts/2015-08-Understanding-LSTMs (дата обращения 14.12.2021).

[3] Там же.

- Временные: время между пакетами (здесь и далее, если возможно: минимальное, максимальное, среднее, в определённом направлении, стандартное отклонение), длительность сессии и т.д.;
- Байтовые: число байтов в потоке, число байтов в заголовках и т. д.
- Пакетные: число пакетов в потоке, длина пакетов в потоке и т.д.;
- Потоковые: длина потока, протокол;
- Флаговые: число пакетов с определённым TCP-флагом (FIN, SYN, RST, PUSH, URG и т. д.);
- Инициализации шифрования: набор шифров, особенности сертификата и т. д.

80+ признаков потока из pcap-файла выделяет инструмент CICFlowMeter. Также данные можно извлечь из Zeek-логов. Zeek (ранее Bro) – это анализатор трафика, который записывает в отдельные логи информацию по разным протоколам и сетевым событиям (conn.log, http.log, x509.log, tls.log и т.д.) и идентифицирует аномальную активность[4].

В данной работе учитываются длина пакета в байтах и временной промежуток между пакетами. Релевантность этих признаков при исследовании вредоносной активности отмечается в статье [6], которая лежит в основе промышленного инструмента для анализа зашифрованного трафика Cisco ETA (Рис. 4).



Р и с. 4. Длины пакетов и временные промежутки между ними – для запроса Google и активности вируса Bestrafera

F i g. 4. Packet lengths and time intervals between them - for Google query and Bestrafera virus activity

Наиболее полные обзоры исследований обнаружения вредоносной активности, преимущественно с помощью метаданных, а также публичных датасетов сделаны в статьях [7], [8]. Отметим: даже если для датасета опубликованы сырые данные (pcap-файлы), исследователи, как правило, выделяют из них усреднённые по сессии признаки и работают с ними [21-25]. Предложенный в данной работе подход, рассматривающий потоки как временные ряды статистических метаданных, не распространён.

Зашифрованный трафик как совокупность временных рядов

Автором был разработан подход, в котором трафик представляется как совокупность временных рядов. Исходные данные – это сырой трафик, состоящий из пакетов, то есть pcap-файл. Предлагается представить его в виде временных рядов – потоков, то есть соединений между 2 хостами.

Идентификатор потока – пятёрка *(IP_1, IP_2, $port_1$, $port_2$, protocol)*. Ему сооветствует ряд точек вида: *(Δt, length, direction, protocol)*, где:

Δt – промежуток времени от доставки предыдущего пакета в потоке,

length – размер пакета в байтах,

direction – направление пакета: от рассматриваемого хоста или к нему,

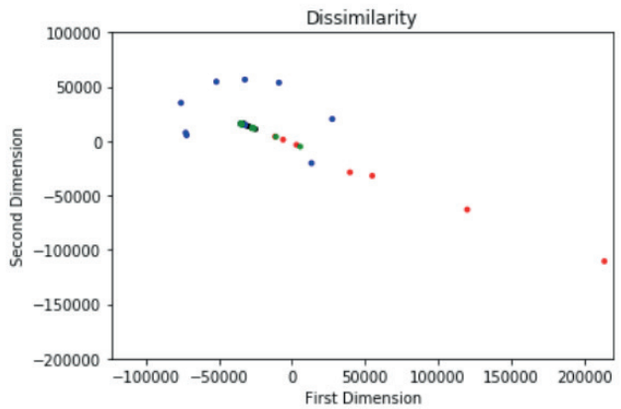
protocol – протокол (его прямое кодирование – one-hot encoding); для зашифрованного трафика выводы о протоколе прикладного уровня можно сделать по TCP-портам.

Ряды приведены к длине не более 30 пакетов для адекватного времени срабатывания классификатора.

На примере реальной сетевой активности вируса-шифровальщика был проведён эксперимент. На полученных временных рядах обучались архитектуры Simple RNN, LSTM, CNN. В качестве базовой модели (baseline) был принят более простой и распространённый подход, в котором показатели пакетов усредняются по потоку; к ним применялись традиционные методы Logistic Regression, RandomForest, SVM, XGBoost. Этот подход показал значительно худшие метрики (Accuracy, Precision, Recall, F-score). Наилучшие результаты показала архитектура LSTM: Accuracy = 0.84, Precision = 0.79, Recall = 0.95, F-score = 0.86.

Был опробован метод обнаружения выбросов в смешанном трафике: DBSCAN по метрике DTW с расстоянием Говера. Чтобы оценить полученную кластеризацию, точки были отображены в 2-мерное пространстве методом многомерного шкалирования (Multidimensional Scaling, MDS) и на график были нанесены нормальные потоки и потоки заражённого хоста без пользовательской активности. Метод показал плохо интерпретируемые результаты (Рис. 5).

[4] Log Files. The Zeek Project, 2021. [Электронный ресурс]. URL: https://docs.zeek.org/en/master/script-reference/log-files.html (дата обращения 14.12.2021).



Р и с. 5. Метрическая визуализация трафика: нормальный трафик – синий цвет; смешанный трафик — чёрный и красный (выбросы); вредоносный трафик – зелёный

F i g. 5. Metric traffic visualization: normal traffic - blue; mixed traffic - black and red (outliers); malicious traffic – green

От классификатора потоков — к классификатору трафика

На основе классификатора потоков построим классификатор трафика , который принимает значение 1, если доля обнаруженных вредоносных потоков в трафике больше определённого порога .

Определение 1. *Если X является дискретной случайной величиной, принимающей неотрицательные целочисленные значения* 0,1,…, *производящая функция вероятностей от*ς*-случайной величины X определяется как*

$$G_X(s) = M(s^X) = \sum_{k=0}^{\infty} P\,(X = k)s^k (1)$$

Для суммы независимых случайных величин $Z = X_1 + \ldots + X_n$

$$G_Z(s) = \prod_{i=1}^{n} G_{X_i}\,(s) (2)$$

Степенной ряд сходится в окрестности 0 (по крайней мере, для $|s| \le 1$). Применима формула Маклорена:

$$P(X = m) = \frac{1}{m!}\frac{d^m G_X(0)}{ds^m} (3)$$

Точность классификатора. Выбор порогового значения
Пусть

α – желаемый порог точности,

событие inf – «хост находится в процессе заражения»,

событие $\overline{inf}$ – «хост не находится в процессе заражения»,

$s_1 \ldots s_n$ – потоки,

θ — доля вредоносных потоков в активном трафике заражённого хоста,

$t = \lfloor \theta n \rfloor$ – количество вредоносных потоков для заражённого хоста,

$c(s_i)$ – классификатор потока,

$C(s_1 \ldots s_n)$ – классификатор трафика, который принимает значение 1, если доля $\{i: c(s_i) = 1\}$ больше определённого порога σ,

$P(TP) = \frac{TP}{TP+FN}$ для классификатора потоков, и так далее.

По формуле полной вероятности выразим точность классификатора трафика:

$$Accuracy = P(inf)P\big(C(s_1 \ldots s_n) = 1|inf\big) + P\big(\overline{inf}\big)P\big(C(s_1 \ldots s_n) = 0|\overline{inf}\big) \ge \alpha$$

В первом слагаемом

$P(inf)$ (a) – доля заражённых хостов от общего количества компьютеров;

$$P\left(C(s_1 \ldots s_n) = 1|inf\right) = P\left(\sum_{i=1}^{n} c\,(s_i) \ge \sigma n \middle| inf\right) = \ldots$$

(замена: $k := \lceil \sigma n \rceil, Z := \sum_{i=1}^{n} c\,(s_i)$)

$$\ldots = P(Z \ge k) = \sum_{m=k}^{n} P\,(Z = m) = \{(3)\} = \sum_{m=k}^{n} \frac{1}{m!}\frac{d^m G_Z(0)}{ds^m} = \ldots$$

$\big(G_Z(s) = \{(2)\} = \prod_{i=1}^{n} G_{c(s_i)}\,(s) = \{(1)\} = \big[P(TP)s + P(FN)\big)^t \big[P(FP)s + P(TN)\big)^{n-t}$;

воспользуемся биномом Ньютона;

заметим, что $\frac{d^m(a_n s^n + \ldots + a_0)}{ds^m}(0) = m!\,a_m\big)$

$\ldots =$
$\sum_{m=k}^{n} \sum_{i=0}^{m} C_t^i\, C_{n-t}^{m-i} P(TP)^i P(FN)^{t-i} P(FP)^{m-i} P(TN)^{n-t-m+i}$. (b)

Аналогично, во втором слагаемом

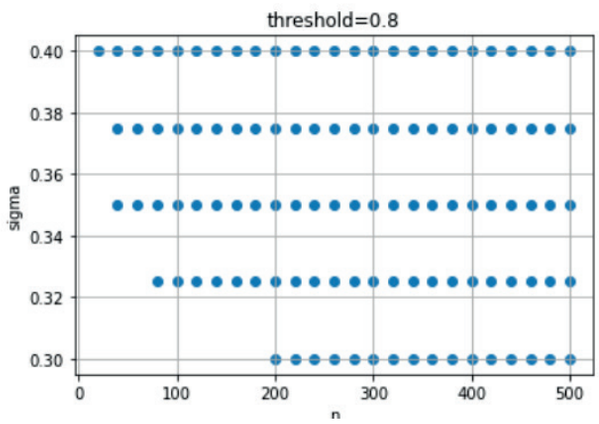
$$P\big(\overline{inf}\big) = 1 - P(inf);\ (c)$$

$$P\big(C(s_1 \ldots s_n) = 0|\overline{inf}\big) = P\big(\sum_{i=1}^{n} c\,(s_i) < \sigma n|\overline{inf}\big) = \ldots = \sum_{m=0}^{k-1} C_n^m\, P(FP)^m P(TN)^{n-m}.\ (d)$$

Таким образом,

$Accuracy(n; \sigma) =$ (a)(b)+(c)(d), где$k = \sigma n, t = \theta n$.

Зафиксировав показатели для классификатора c, можно построить сетку значений (n; σ) и определить точки, в которых точность классификатора C(*Accuracy*) не меньше заданного порога α (Рис. 6). Это позволит, зафиксировав количество доступных потоков n, выбирать пороговое значение σ.



Р и с. 6. Сетка значений (n; σ) для проведённого эксперимента, при заданном пороге точности α = 0.8: синим отмечены точки, в которых достигается не меньшая точность. Доля вредоносных потоков θ = 0.03; для достижения высокой точности значение σ должно быть намного больше из-за погрешностей классификации

F i g. 6. Value grid (n; σ) for the experiment, with a given threshold of accuracy α = 0.8: blue marks the points at which the same accuracy is achieved. Percentage of malicious threads θ = 0.03; to achieve high accuracy σ, the value should be much larger due to classification errors

Заключение

Подход к решению задачи, в котором трафик рассматривается как совокупность временных рядов, показал удовлетворительные результаты даже при рассмотрении всего 4 признаков и в условиях ненадёжно размеченного трафика. Он превзошёл более распространённый подход, в котором данные потока усредняются. На основании метрик качества можно выделить архитектуру LSTM.

На основе детектора вредоносных потоков смоделирован детектор вредоносного трафика, и выведена формула его точности от количества потоков и порога принятия решения. Эта модель применима для любой классификации, в которой за единицу трафика принят поток и рассматривается смешанный трафик с фиксированной долей вредоносных потоков.

Перспектива для дальнейших исследований – многоклассовая классификация с несколькими типами вредоносного трафика, а также проблема атаки на обученную модель, то есть обхода детектора.

References

[1] Lakshmanarao A., Shashi M. A Survey On Machine Learning For Cyber Security. *International Journal of Scientific & Technology Research*. 2020; 9(01):499-502. Available at: https://www.ijstr.org/final-print/jan2020/-A-Survey-On-Machine-Learning-For-Cyber-Security.pdf (accessed 14.12.2021). (In Eng.)

[2] Susto G.A., Cenedese A., Terzi M. Chapter 9: Time-Series Classification Methods: Review and Applications to Power Systems Data. In: Arghandeh R., Zhou Y. (eds.). *Big Data Application in Power Systems*. Elsevier Inc.; 2018. p. 179-220. (In Eng.) doi: https://doi.org/10.1016/B978-0-12-811968-6.00009-7

[3] Wang W., Zhu M., Zeng X., Ye X., Sheng Y. Malware traffic classification using convolutional neural network for representation learning. *2017 International Conference on Information Networking (ICOIN).* IEEE Press, Da Nang, Vietnam; 2017. p. 712-717. (In Eng.) doi: https://doi.org/10.1109/ICOIN.2017.7899588

[4] Alom Z., Bontupalli V.R., Taha T.M. Intrusion Detection Using Deep Belief Network and Extreme Learning Machine. In: *Artificial Intelligence: Concepts, Methodologies, Tools, and Applications*; ed. by Management Association, Information Resources. Hershey, PA: IGI Global; 2017. p. 357-378. (In Eng.) doi: https://doi.org/10.4018/978-1-5225-1759-7.ch014

[5] Kim Ji., Kim Ja., Thi Thu H.L., Kim H. Long Short Term Memory Recurrent Neural Network Classifier for Intrusion Detection. *2016 International Conference on Platform Technology and Service (PlatCon).* IEEE Press, Jeju, Korea (South); 2016. p. 1-5. (In Eng.) doi: https://doi.org/10.1109/PlatCon.2016.7456805

[6] Anderson B., McGrew D. Identifying Encrypted Malware Traffic with Contextual Flow Data. *Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security (AISec'16).* Association for Computing Machinery, New York, NY, USA; 2016. p. 35-46. (In Eng.) doi: https://doi.org/10.1145/2996758.2996768

[7] Ahmad Z., Khan A.S., Shiang C., Ahmad F. Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies*. 2021; 32(1):e4150. (In Eng.) doi: https://doi.org/10.1002/ett.4150

[8] Wang Z., Fok K.W., Thing V.L.L. Machine learning for encrypted malicious traffic detection: Approaches, datasets and comparative study. *Computers and Security*. 2022; 113(C). 22 p. (In Eng.) doi: https://doi.org/10.1016/j.cose.2021.102542

[9] Anderson B., Paul S., McGrew D. Deciphering malware's use of TLS (without decryption). *Journal of Computer Virology and Hacking Techniques.* 2018; 14(3):195-211. (In Eng.) doi: https://doi.org/10.1007/s11416-017-0306-6

[10] Gentry C., Halevi S. Implementing Gentry's Fully-Homomorphic Encryption Scheme. In: Paterson K.G. (ed.) *Advances in Cryptology – EUROCRYPT 2011. EUROCRYPT 2011. Lecture Notes in Computer Science.* Vol. 6632. Springer, Berlin, Heidelberg; 2011. p. 129-148. (In Eng.) doi: https://doi.org/10.1007/978-3-642-20465-4_9

[11] Brakerski Z., Gentry C., Vaikuntanathan V. (Leveled) Fully Homomorphic Encryption without Bootstrapping. *ACM Transactions on Computation Theory*. 2014; 6(3):13. (In Eng.) doi: https://doi.org/10.1145/2633600

[12] Kang H., Lee D.H. Security Assessment for Application Network Services Using Fault Injection. In: Yang C.C., et al. (eds.). *Intelligence*

and Security Informatics. PAISI 2007. Lecture Notes in Computer Science. Vol. 4430. Springer, Berlin, Heidelberg; 2007. p. 172-183. (In Eng.) doi: https://doi.org/10.1007/978-3-540-71549-8_15

[13] Domingo-Ferrer J. A Provably Secure Additive and Multiplicative Privacy Homomorphism. In: Chan A.H., Gligor V. (eds.). *Information Security. ISC 2002. Lecture Notes in Computer Science*. Vol. 2433. Springer, Berlin, Heidelberg; 2002. p. 471-483. (In Eng.) doi: https://doi.org/10.1007/3-540-45811-5_37

[14] Yao A.C. Protocols for secure computations. *23rd Annual Symposium on Foundations of Computer Science (sfcs 1982).* IEEE Press, Chicago, IL, USA; 1982. p. 160-164. (In Eng.) doi: https://doi.org/10.1109/SFCS.1982.38

[15] Goldreich O., Micali S., Wigderson A. How to play ANY mental game. *Proceedings of the nineteenth annual ACM symposium on Theory of computing (STOC'87).* Association for Computing Machinery, New York, NY, USA; 1987. p. 218-229. (In Eng.) doi: https://doi.org/10.1145/28395.28420

[16] Mohassel P., Zhang Y. SecureML: A System for Scalable Privacy-Preserving Machine Learning. *2017 IEEE Symposium on Security and Privacy (SP).* IEEE Press, San Jose, CA, USA; 2017. p. 19-38. (In Eng.) doi: https://doi.org/10.1109/SP.2017.12

[17] Peikert C., Vaikuntanathan V., Waters B. A Framework for Efficient and Composable Oblivious Transfer. In: Wagner D. (ed.). *Advances in Cryptology – CRYPTO 2008. CRYPTO 2008. Lecture Notes in Computer Science.* Vol. 5157. Springer, Berlin, Heidelberg; 2008. p. 554-571. (In Eng.) doi: https://doi.org/10.1007/978-3-540-85174-5_31

[18] Niksefat S., Sadeghiyan B., Mohassel P., Sadeghian S. ZIDS: A Privacy-Preserving Intrusion Detection System Using Secure Two-Party Computation Protocols. *The Computer Journal.* 2014; 57(4):494-509. (In Eng.) doi: https://doi.org/10.1093/comjnl/bxt019

[19] Alhassan M.Y., Günther D., Kiss Á., et al. Efficient and Scalable Universal Circuits. *Journal of Cryptology*. 2020; 33(3):1216-1271. (In Eng.) doi: https://doi.org/10.1007/s00145-020-09346-z

[20] Canetti R. Universally composable security: a new paradigm for cryptographic protocols. *Proceedings 42nd IEEE Symposium on Foundations of Computer Science.* IEEE Press, Newport Beach, CA, USA; 2001. p. 136-145. (In Eng.) doi: https://doi.org/10.1109/SFCS.2001.959888

[21] Damgard I., Geisler M., Kroigard M. Homomorphic encryption and secure comparison. *International Journal of Applied Cryptography.* 2008; 1(1):22-31. Available at: https://www.inderscienceonline.com/doi/abs/10.1504/IJACT.2008.017048 (accessed 14.12.2021). (In Eng.)

[22] Gilad-Bachrach R., Dowlin N., Laine K., Lauter K., Naehrig M., Wernsing J. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. *Proceedings of the 33rd International Conference on Machine Learning (PMLR).* New York, NY, USA. 2016; 48:201-210. Available at: https://proceedings.mlr.press/v48/gilad-bachrach16.html (accessed 14.12.2021). (In Eng.)

[23] Dhote Y., Agrawal S., Deen A.J. A Survey on Feature Selection Techniques for Internet Traffic Classification. *2015 International Conference on Computational Intelligence and Communication Networks (CICN).* IEEE Press, Jabalpur, India; 2015. p. 1375-1380. (In Eng.) doi: https://doi.org/10.1109/CICN.2015.267

[24] Gao N., Gao L., Gao Q., Wang H. An Intrusion Detection Model Based on Deep Belief Networks. *2014 Second International Conference on Advanced Cloud and Big Data.* IEEE Press, Huangshan, China; 2014. p. 247-252. (In Eng.) doi: https://doi.org/10.1109/CBD.2014.41

[25] Koukis D., Antonatos S., Antoniades D., Markatos E.P., Trimintzios P. A Generic Anonymization Framework for Network Traffic. *2006 IEEE International Conference on Communications*. IEEE Press, Istanbul, Turkey; 2006. p. 2302-2309. (In Eng.) doi: https://doi.org/10.1109/ICC.2006.255113

Поступила 14.12.2021; одобрена после рецензирования 25.02.2022; принята к публикации 04.03.2022.
Submitted 14.12.2021; approved after reviewing 25.02.2022; accepted for publication 04.03.2022.

Об авторе:

Полянская Марина Сергеевна, магистрант кафедры информационной безопасности, факультет вычислительной математики и кибернетики, ФГБОУ ВО «Московский государственный университет имени М. В. Ломоносова» (119991, Российская Федерация, г. Москва, ГСП-1, Ленинские горы, д. 1), **ORCID: https://orcid.org/0000-0002-6848-0413,** m.s.polyanskaya@mail.ru

Автор прочитал и одобрил окончательный вариант рукописи.

About the author:

Marina S. Polyanskaya, Master degree student of the Chair of Information Security, Faculty of Computational Mathematics and Cybernetics, Lomonosov Moscow State University (1 Leninskie gory, Moscow 119991, GSP-1, Russian Federation), **ORCID: https://orcid.org/0000-0002-6848-0413,** m.s.polyanskaya@mail.ru

The author has read and approved the final manuscript.