Антон А. Недогарок[1], Николай В. Федоров[2], Виталий С. Швычков[3], Максим И. Калайда[4]
[1] Московский государственный технический университет им. Н.Э. Баумана,
2-я Бауманская ул., 5, стр.1, Москва, 105005, Россия
[1]e-mail: nedogarok.aa@bmstu.ru, https://orcid.org/0000-0002-6146-674X
[2,3,4]Московский политехнический университет,
Большая Семеновская ул., 38, Москва, 107023, Россия
[2]e-mail: fedorovNV31@mail.ru, https://orcid.org/0000-0002-3362-144X
[3]e-mail: 414kappat@gmail.com, https://orcid.org/0000-0003-3618-4086
[4]e-mail: maxkalayda@gmail.com, https://orcid.org/0000-0002-7245-8235

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МОДУЛЯ DLP-СИСТЕМЫ ДЛЯ МОНИТОРИНГА И АНАЛИЗА ТРАФИКА КОРПОРАТИВНОЙ СЕТИ С ИСПОЛЬЗОВАНИЕМ МАШИННОГО ОБУЧЕНИЯ

DOI: http://dx.doi.org/10.26583/bit.2020.1.03

Аннотация. Предметом работы является нейронная сеть, параметры её архитектуры и массив данных (датасет) для её обучения. Целью работы является программная реализация части DLP-системы, позволяющей осуществлять мониторинг трафика корпоративной сети и контролировать пересылку конфиденциальных данных по этой сети при помощи нейронной сети. Весь процесс разработки представлен пятью этапами: теория, проектирование, подготовка данных для обучения нейронной сети, обучение нейронной сети и тестирование реализованной системы. Приведён краткий обзор рынка подобных решений. Подробно описаны используемые параметры для построения архитектуры нейронной сети, используемой для решения задачи классификации текстовых данных. Результатом работы является функционирующая часть DLP-системы, позволяющая осуществлять мониторинг трафика корпоративной сети через веб-интерфейс и контролировать пересылку конфиденциальных данных по этой сети при помощи одномерной свёрточной нейронной сети 1D CNN.

Ключевые слова: DLP-система, машинное обучение, нейронная сеть, конфиденциальная информация, классификация текста.

Для цитирования: НЕДОГАРОК, Антон А. et al. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МОДУЛЯ DLP-СИСТЕМЫ ДЛЯ МОНИТОРИНГА И КОНТРОЛЯ ТРАФИКА КОРПОРАТИВНОЙ СЕТИ С ИСПОЛЬЗОВАНИЕМ МАШИННОГО ОБУЧЕНИЯ. Безопасность информационных технологий, [S.l.], v. 27, n. 1, p. 28-39, 2020. ISSN 2074-7136. Доступно на: <https://bit.mephi.ru/index.php/bit/article/view/1252>. Дата обращения: 11 feb. 2020. DOI: http://dx.doi.org/10.26583/bit.2020.1.03.

Anton A. Nedogarok[1], Nikolai V. Fedorov[2], Vitaly S. Shvychkov[3], Maxim I. Kalayda[4]
[1]Bauman Moscow State Technical University,
2nd Baumanskaya St., 5, bld. 1, Moscow, 105005, Russia
[1]e-mail: nedogarok.aa@bmstu.ru, https://orcid.org/0000-0002-6146-674X
[2,3,4]Moscow Polytechnical University,
Bolshaya Semenovskaya St., 38, Moscow, 10702, Russia
[2]e-mail: fedorovNV31@mail.ru, https://orcid.org/0000-0002-3362-144X
[3]e-mail: 414kappat@gmail.com, https://orcid.org/0000-0003-3618-4086
[4]e-mail: maxkalayda@gmail.com, https://orcid.org/0000-0002-7245-8235

Software implementation of a DLP-system module for monitoring and analysis corporate network traffic using machine learning

DOI: http://dx.doi.org/10.26583/bit.2020.1.03

Abstract. The subject of this work is a neural network, the parameters of its architecture and the data array (dataset) for its training. The aim of the work is the software implementation of part of the DLP-system (Data Leak Prevention), which allows to monitor the traffic of a corporate network and to control the transfer of confidential data over this network using a neural network. The entire development process is

represented by five stages: theory, design, preparation of data for training the neural network, training the neural network and testing the implemented system. There is a brief overview of the market for such solutions in the article. The parameters used to construct the neural network architecture used to solve the problem of text data classification are described in detail. The result of the work is a functioning part of the DLP system, which allows monitoring the traffic of a corporate network via a web-interface and controlling the transfer of confidential data over this network using a one-dimensional convolutional neural network 1D CNN.

Keywords: DLP-system, machine learning, neural network, confidential information, text classification, dataset.

For citation: NEDOGAROK, Anton A. et al. Software implementation of a DLP-system module for monitoring and controlling corporate network traffic using machine learning. IT Security (Russia), [S.l.], v. 27, n. 1, p. 28-39, 2020. ISSN 2074-7136. Available at: <https://bit.mephi.ru/index.php/bit/article/view/1252>. Date accessed: 11 feb. 2020. DOI: http://dx.doi.org/10.26583/bit.2020.1.03.

Введение

Количество обрабатываемых данных с каждым годом растёт, как для крупных компаний, так и для небольших организаций, а для передачи информации преимущественно используется корпоративная сеть и/или сеть Интернет. Обмен в сети данными, содержащими конфиденциальную информацию, должен контролироваться для предотвращения утечек такой информации. Каналы утечки информации в сети могут быть использованы злоумышленником, чтобы передать конфиденциальные данные за пределы организации. Пример такого канала утечки: протокол доставки электронной почты SMTP (Simple Mail Transfer Protocol). Злоумышленник может прикрепить к письму конфиденциальный документ, отправить его на внешний электронный адрес и получить доступ к похищенным данным за территорией организации. Далее на примере перехвата и проверки SMTP-трафика будет рассмотрен пример системы мониторинга информационной безопасности сетей/систем.

Для предотвращения реализации подобных угроз следует использовать программные и программно-аппаратные средства защиты, предоставляющие возможность контролировать доступ внутри корпоративной сети и к ресурсам Интернета. Средства подобного типа должны выявлять и блокировать попытки несанкционированной передачи конфиденциальной информации по сетевым каналам утечки в организации. Одним из таких средств предотвращения несанкционированной передачи конфиденциальной информации по сети является DLP-система [1].

В современных системах предотвращения утечек конфиденциальной информации используются дорогие, трудно масштабируемые и ресурсоёмкие решения, иногда требующие специальный отдел сотрудников, занимающихся обслуживанием DLP-системы. Для автоматизации и оптимизации работы таких систем используется технология машинного обучения, а именно – нейронные сети.

На рынке присутствует широкий выбор DLP-систем для выполнения большого набора задач в различных сетях и системах. В табл. 1 приведены примеры популярных и ведущих, по мнению аналитиков исследовательской и консультационной компании Gartner, представителей рынка DLP-систем по состоянию на 2018 год [2].

Из табл. 1 можно сделать вывод, что современные DLP-системы используют машинное обучение для увеличения скорости и точности обнаружения конфиденциальной информации в потоке данных, для работы с большими объёмами труднообрабатываемых данных и для других целей [3–11].

Таблица 1. Сравнительная таблица представителей рынка DLP-систем

Название продукта	McAfee Total Protection for DLP	Symantec DLP	Triton DLP	InfoWatch Traffic Monitor
Логотип	McAfee	Symantec	FORCEPOINT	INFOWATCH
Страна производитель	США	США	США	Россия
Контроль Web-сервисов и приложений	Да (McAfee DLP Prevent)	Да	Да (Web DLP модуль)	Да
Использование машинного обучения	Да (Big Data)	Да (классификация данных, OCR)	Да (идентификация конфиденциальной информации)	Нет сведений
Минимальные системные требования	2xHDD 600 GB, 2xCPU Intel Xeon E5-2620 V4 8xCores 2.1 GHz, 8x4Gb 2400Mhz RAM.	Нет сведений	Нет сведений	232GB HDD, 1xCPU 6xCores 2,6 Ghz, 8 GB RAM, 1xServer.

1. Проектирование модуля DLP-системы

Искусственный интеллект, нейронные сети и машинное обучение – это разные понятия [12]. Их отличия (отношения друг к другу) представлены на рис. 1.

Рис. 1. Схема отношений искусственного интеллекта, машинного обучения и искусственных нейронных сетей друг к другу
(Fig. 1. The relationship between artificial intelligence, machine learning and artificial neural networks to each other)

Существуют различные архитектуры нейронных сетей [12]. В данной статье для проектируемого модуля DLP-системы используется архитектура одномерной свёрточной нейронной сети 1D CNN (Convolutional Neural Network) (рис. 2). Согласно проведённому обзору различных источников информации и примеров практического использования различных архитектур нейронных сетей [13], эта архитектура часто используется для решения задач классификации текстовых данных. Поэтому сеть 1D CNN подходит для обработки содержимого электронных писем, их заголовков и прикреплённых файлов. Основное преимущество архитектуры 1D CNN – высокая скорость обучения и работы нейронной сети.

Архитектура 1D CNN (см. рис. 2) реализована на языке программирования Python 3.7 [14] с использование фреймворков Keras [15] и Tensorflow [16]. Модель реализована на основе последовательной (Sequential) структуры слоёв, предоставляющей удобный интерфейс для работы со структурой. Последующие попытки обучения и тестирование обученных моделей нейронной сети позволило выявить наилучшее сочетание параметров для данной архитектуры.

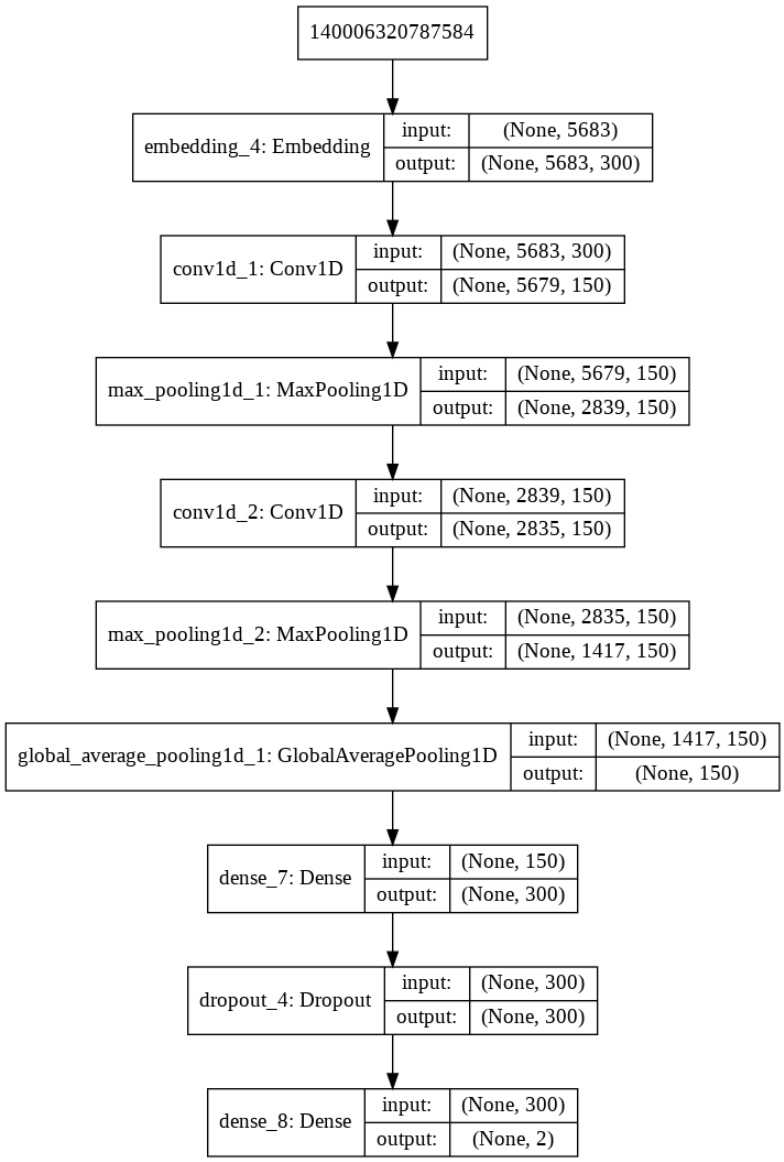


Рис. 2. Схема архитектуры 1D CNN
(Fig. 2. Architecture scheme of the 1D CNN)

Первый слой – входной слой (Embedding) содержащий следующие аргументы:
– «input_dim = vocab_size» – размер словаря;
– «output_dim = emdedding_size» – размерность данных на выходе;
– «input_length = max_length» – длина блока данных на входе;
– «weights = [pretrained_weights]» – заранее подготовленные значения векторов слов.

Второй и четвёртый слои – слои свёртки (Conv1D) с аргументами:
– «filters = 150» – размерность данных на выходе (количество фильтров на выходе свёртки);
– «kernel_size = 5» – размерность ядра свёртки (в данном случае ядро свёртки имеет размерность 5x1);
– «activation = 'relu'» - функция активации (в данном случае функцией активации является ReLU).

Третий и пятый слои – слои субдискретизации (MaxPooling1D). Эти слои имеют следующие аргументы:

– «pool_size = 2» – максимальная размерность окна объединения;
– «strides = pool_size» – длина сдвига после объединения.

Шестой слой – слой общего среднего максимального объединения для временных данных (GlobalAveragePooling1D), не принимающий аргументы.

Седьмой слой – полносвязный слой (Dense) с аргументами:

– «units = 300» – размерность данных на выходе;
– «activation = 'relu'» – функция активации.

Восьмой слой – слой сброса значений весов (Dropout) с аргументом «rate = 0.5». Этот аргумент задаёт вероятность сброса весов (в данном случае вероятность равна 50%).

Девятый слой – ещё один полносвязный слой (Dense), имеющий аргументы:

– «units = 2» – размерность данных на выходе;
– «activation = 'sigmoid'» – функция активации (в данном случае функцией активации является сигмоида).

Таким образом, конечная модель нейронной сети (model) представляет собой объект с линейной последовательностью из 9 слоев, готовый к сборке (компиляции). Процесс компиляции нейронной сети осуществляется с помощью оператора «.compile», имеющего аргументы:

– «loss = 'binary_crossentropy'» – функция ошибки;
– «optimizer = 'adam'» – алгоритм оптимизации;
– «metrics = ['accuracy']» – метрика, используемая при обучении (в данном случае используется «точность»).

Для проектирования модуля системы мониторинга безопасности информационной сети в качестве примера рассматривается модуль перехвата исходящего SMTP-трафика корпоративной сети [17]. В корпоративной сети пользователь отправляет данные, содержащие конфиденциальную информацию, через корпоративный сервер-перехватчик (SMTP-relay) (рис. 3).

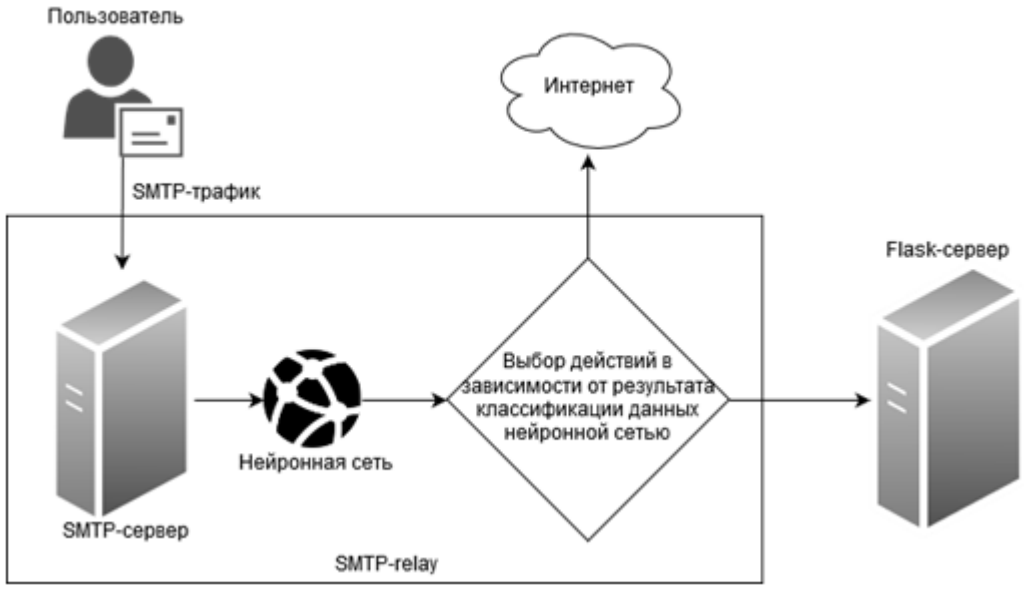


Рис. 3. Схема движения SMTP-трафика через модуль DLP-системы
(Fig. 3. Scheme of movement of SMTP traffic through the module of the DLP system)

На этом сервере находится встроенный модуль DLP-системы, использующий машинное обучение (нейронную сеть). Данные проходят операции преобразования, а затем отправляются SMTP-сервером в нейронную сеть. Нейронная сеть классифицирует полученные данные. Если данные прошли проверку (нейронная сеть не обнаружила признаков конфиденциальной информации), они отправляются дальше по протоколу SMTP во внешнюю сеть. Если данные не прошли проверку (нейронная сеть обнаружила признаки конфиденциальной информации), то далее письмо не отправляется, а содержимое письма сохраняется на Flask-сервер [18] для дальнейшего информирования офицера безопасности, предоставления статистики и другого вида информации.

Для комфортной работы с нейронной сетью и отчётностью на Flask-сервере реализуется web-интерфейс (web-страница) с использованием языков Python, HTML, CSS и JavaScript, в котором отображается основная информация об обнаруженных утечках, т.е. через этот сервер осуществляется мониторинг безопасности информационной сети организации. Пример дизайна веб-страницы модуля DLP-системы представлен на рис. 4.

Рис. 4. Пример дизайна веб-интерфейса модуля DLP-системы
(Fig. 4. An example of the design of the web interface of the DLP system)

Основной информацией об обнаруженных утечках на рис. 4 является: почтовый адрес отправителя письма с конфиденциальной информацией, почтовый адрес получателя письма с конфиденциальной информацией, время перехвата письма модулем DLP-системы, данные о количестве инцидентов в день за заданный промежуток времени. Список отображаемой на web-странице информации об обнаруженных утечках может быть изменён в зависимости от получения требуемой информации в конкретной DLP-системе.

2. Подготовка данных и составление датасета

Для обучения и проверки нейронной сети необходимо иметь датасет, разделённый на необходимые классы: в данном случае это файлы, содержащие и не содержащие конфиденциальную информацию (например, персональные данные). Для решения задачи классификации текстовых документов в качестве примера используются файлы с

расширением «.txt», так как такого формата файлы легко поддаются редактированию сторонними программами и инструментами и имеют малый вес.

Каждый файл датасета формируется из содержимого одного из источников, таких как: интернет-газеты, FTP-сервер единой информационной системы в сфере закупок, свободная энциклопедия, собственный генератор текстовых файлов и другие свободные источники в сети Интернет. Сформированные файлы составляют датасет, содержащий конфиденциальную информацию (примеры паспортных данных, различных идентификационных номеров и т.д.) и не конфиденциальную информацию (литературные тексты, тексты новостных статей и т.д.).

Итоговый объём датасета для обучения нейронной сети составляет более 3000 текстовых файлов с расширением «.txt». Из них 50% файлов содержат конфиденциальную информацию, 50% файлов не содержат конфиденциальную информацию. Конфиденциальные и не конфиденциальные файлы разделены заранее и расположены в отдельных директориях для упрощения реализации обучения нейронной сети. Файлы датасета содержат тексты длиной от 500 до 6500 слов. Значения количества файлов в датасете и количества слов в каждом файле датасета не являются окончательными. В данной статье эти значения ограничены временем сбора информации для датасета и аппаратной мощностью для обучения, тестирования и эксплуатации нейронной сети. При увеличении количества времени, затрачиваемого на составление датасета, и аппаратной мощности для работы с нейронной сетью, возможно увеличение как объёма датасета, так и количества слов в каждом файле.

Такой набор данных в датасете установлен на основе проведённого обзора различных источников информации и примеров практического использования датасета в решении подобных задач [13]. После тестов и нескольких попыток обучить нейронную сеть датасет был скорректирован: сведены к минимуму повторения частей текста в разных файлах, увеличена разница длин текстов в файлах, увеличено количество тем и шаблонов для текстов и т.п. Чем больше уникальных слов в общем словаре всех текстов датасета, чем более разнообразна структура и шаблоны текстов в файлах, чем больше оригинальность содержимого каждого файла датасета, тем датасет более насыщен. При увеличении размера и насыщенности датасета увеличится и точность работы обученной нейронной сети, однако потребуется больше аппаратной мощности для обучения нейронной сети.

После подготовки и разделения файлов датасета, назначения каждому из них своего класса (табл. 2) их текстовое содержимое преобразуется в числовой вид для применения в математических функциях нейронной сети при обучении.

Таблица 2. Пример текстового содержимого файлов датасета и назначенных им классов (0 – не содержит конфиденциальную информацию, 1 – содержит)

X_train	Y_train	X_test	Y_test
Паспортные данные	1	Договор с банком	1
Договор с банком	1	Паспортные данные	1
Сборник стихов	0	Научная статья	0
Научная статья	0	Сборник стихов	0

3. Обучение нейронной сети

Для повышения точности работы нейронной сети необходимо сохранить зависимости слов текста друг от друга. Для решения этой задачи используется технология Word2Vec [19].

В качестве входных данных алгоритм Word2Vec использует текст из файлов датасета. Во время своего «обучения» Word2Vec составляет словарь всех слов текста, после чего для каждого слова составляет вектор с координатами слова, учитывая рядом стоящие слова. Такая векторная запись слов основана на контекстной близости: близкие векторные координаты будут иметь слова, находящиеся в тексте рядом с другими одинаковыми словами.

Программная реализация начинается с создания модели model_w2v для алгоритма Word2Vec со следующими аргументами:

- «x = X_train» – текстовые данные;
- «iter = 100» – количество проходов алгоритма по тексту;
- «min_count = 1» – минимальное количество встречи слова, чтобы оно было занесено в словарь;
- «window = 5» – количество слов, стоящих рядом с текущим словом, связи с которыми будут учитываться при построении вектора для текущего слова;
- «size = 300» – длина выходного вектора;
- «workers = 4» – количество потоков процессора, которые могут одновременно работать с Word2Vec.

После исполнения алгоритма «Word2Vec()» с перечисленными выше аргументами для получения векторов слов (pretrained_weights) необходимо применить метод «.wv.syn0» на model_w2v. Размерность pretrained_weights характеризуется размером словаря слов из всех текстов (vocab_size) и размером выходного вектора (параметр «size») для входного слоя нейронной сети (emdedding_size).

Итогом работы Word2Vec являются сформированные на основе содержания текстов вектора слов, размер словаря всех слов из всех текстов и размер выходного вектора для входного слоя нейронной сети.

Далее каждому уникальному слову из датасета присваивается индекс, подсчитывается размер словаря датасета, и слова преобразуются в последовательность чисел типа integer с использованием методов класса «Tokenizer» [20].

Для использования Tokenizer’a создаётся объект класса «tokenizer_obj = Tokenizer()», в который при помощи метода «.fit_on_texts» помещается X_train и формируется словарь всех слов всех текстов, входящих в X_train.

При помощи функции «max()» определяется количество символов (max_length) в самой большой последовательности символов, содержащейся в X_train.

Далее X_train и X_test как аргументы для метода «tokenizer_obj.texts_to_sequences», преобразуются в числовые последовательности типа integer.

После применяется «sequence.pad_sequences()» с аргументами:

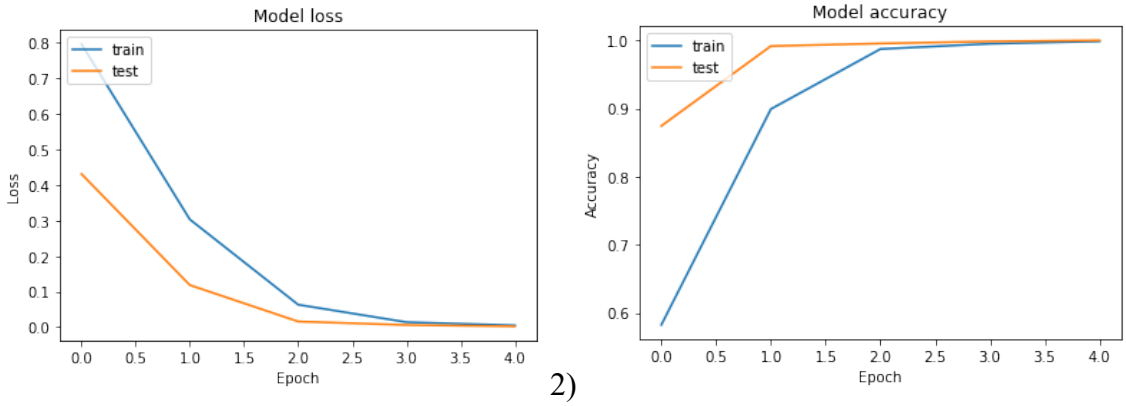
- «X_train» или «X_test» – обрабатываемые списки числовых последовательностей;
- «padding = 'post'» – добавление заданных чисел в конец числовой последовательности для доведения длины последовательности до значения max_length;
- «maxlen = max_length» – длина, до которой будет наращиваться числовая последовательность;
- «value = 0.0» – число, которое будет использоваться для заполнения недостающей части числовой последовательности.

Таким образом, X_train и X_test из списков последовательностей текстовых данных преобразуются в списки числовых последовательностей одинаковой длины.

Здесь же Y_train и Y_test преобразуются в числовые последовательности типа float16 при помощи «utils.to_categorical()» с аргументами:

– «Y_train» или «Y_test» – обрабатываемые списки числовых последовательностей;
– «num_classes = 2» – количество используемых классов;
– «dtype = 'float16'» – преобразование числовой последовательности к типу данных float16.

Далее уже числовые данные помещаются в нейронную сеть, после чего запускается процесс обучения. Нейронная сеть «запоминает» какой текст к какому классу относится, меняя между нейронами веса после заданного количества прочтённых файлов с их классами. Период между изменениями весов называется «эпохой», а набор читаемых за эпоху файлов – «партией». С каждой эпохой нейронная сеть становится точнее в решении поставленной задачи, и в конце обучения получается набор нейронов, связанных синапсами с весами, изменёнными на основании рассмотренных входных данных (рис. 5).



1) 2)

Рис. 5. Графики работы нейронной сети:
1) с уменьшением процента ошибок классификации данных во время обучения,
2) с увеличением процента правильной классификации данных во время обучения
(Fig. 5. The schedules of the neural network:
1) with a decrease in the percentage of data classification errors during training,
2) with an increase in the percentage of correct data classification during training)

Обученная нейронная сеть представляет из себя файл, который помещается на сервер-перехватчик и подключается через инструменты Flask к функционалу сервера.

Например, для выбранной архитектуры 1D CNN (рис. 2) были выбраны значения epochs (эпох) = 5, a batch_size (партия) = 300. Процесс обучения при таких параметрах приведён на рис. 6.

```
Train on 2644 samples, validate on 661 samples
Epoch 1/5
2644/2644 [=========] - 25s 9ms/step - loss: 0.8412 - acc: 0.5739 - val_loss: 0.4674 - val_acc: 0.8933
Epoch 2/5
2644/2644 [=========] - 10s 4ms/step - loss: 0.3252 - acc: 0.8816 - val_loss: 0.1177 - val_acc: 0.9856
Epoch 3/5
2644/2644 [=========] - 10s 4ms/step - loss: 0.0663 - acc: 0.9854 - val_loss: 0.0206 - val_acc: 0.9909
Epoch 4/5
2644/2644 [=========] - 10s 4ms/step - loss: 0.0174 - acc: 0.9945 - val_loss: 0.0056 - val_acc: 0.9977
Epoch 5/5
2644/2644 [=========] - 10s 4ms/step - loss: 0.0043 - acc: 0.9992 - val_loss: 0.0023 - val_acc: 1.0000
```

Рис. 6. Отображение процесса обучения 1D CNN
(Fig. 6. Display of the 1N CNN learning process)

4. Тестирование обученной нейронной сети

В процессе тестирования так же, как и во время обучения, назначается размер партии (batch_size) работы для нейронной сети (рис. 7). Важно использовать для теста исходные данные, которые не входят в датасет, использованный при обучении нейронной сети. Это позволит провести независимую оценку работы нейронной сети. Независимый датасет составлен из десяти текстовых файлов, не содержащих конфиденциальную информацию (обозначаются цифрой «0»), и десяти текстовых файлов, содержащих конфиденциальную информацию (обозначаются цифрой «1»). Файлы помещены в датасет в этом же порядке. То есть верная классификация текстов файлов независимого датасета (набор элементов списка «Classes» на рис. 7) выглядит так: 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1.

Для решения задач классификации текстовых данных кроме одномерной сети 1D CNN часто используют LSTM-нейронные сети (Long Short-Term Memory). Однако по результатам тестирования модели LSTM-нейронной сети и подбора различных параметров, для обучения LSTM-нейронной сети, получен результат с низкой точностью работы (рис. 7), а процессы обучения и работы нейронной сети с такой архитектурой занимают в разы больше времени, чем 1D CNN. Поэтому, сеть 1D CNN используется в качестве основной архитектуры для нейронной сети в итоговом модуле системы.

1)

```
classes = model.predict_classes(data_eat, batch_size=20)
print('Classes = ', classes)
proba = model.predict_proba(data_eat, batch_size=20)
print('Proba = ', proba)
```

```
Classes =  [0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1]
Proba =  [[0.9929086  0.00797361]
 [0.8326957  0.14015627]
 [0.864302   0.11100486]
 [0.84336054 0.12882292]
 [0.8204459  0.14961502]
 [0.7705933  0.19822404]
 [0.7910508  0.17766568]
 [0.91285217 0.08084452]
 [0.75649786 0.19849285]
 [0.97891295 0.02181676]
 [0.0234001  0.9626507 ]
 [0.01785994 0.9707114 ]
 [0.16514671 0.75352466]
 [0.42787817 0.46652552]
 [0.06803343 0.8775927 ]
 [0.21799263 0.6842015 ]
 [0.07178655 0.8744367 ]
 [0.13420272 0.7970988 ]
 [0.02501947 0.9544934 ]
 [0.18841943 0.7176654 ]]
```

2)

```
classes = model.predict_classes(data_eat, batch_size=20)
print('Classes = ', classes)
proba = model.predict_proba(data_eat, batch_size=20)
print('Proba = ', proba)
```

```
Classes =  [0 0 0 0 0 1 0 0 0 0 1 1 0 0 0 0 0 1 0 0]
Proba =  [[9.9892688e-01 1.0603964e-03]
 [9.9733984e-01 2.8394759e-03]
 [9.9868834e-01 1.2687743e-03]
 [9.9717832e-01 2.7173460e-03]
 [9.9810064e-01 1.9031763e-03]
 [4.0820009e-01 6.2625819e-01]
 [9.8303711e-01 1.6585797e-02]
 [9.9813557e-01 1.8010437e-03]
 [9.9888831e-01 1.0976493e-03]
 [9.9913120e-01 8.6721778e-04]
 [2.6383013e-02 9.7397190e-01]
 [2.6383013e-02 9.7397190e-01]
 [9.9685615e-01 3.4874678e-03]
 [9.9825418e-01 1.7383993e-03]
 [8.9324707e-01 1.0335204e-01]
 [9.9839002e-01 1.6236603e-03]
 [7.0541203e-01 3.1976867e-01]
 [3.6362052e-02 9.6547019e-01]
 [6.8211919e-01 3.1086129e-01]
 [9.9577343e-01 4.2683482e-03]]
```

Рис. 7. Результаты проверки независимого датасета из двадцати файлов обученными нейронными сетями (Classes – класс файла «по мнению» нейронной сети, Proba – процент соответствия содержимого файла классу (слева – 0, справа - 1)): 1) 1D CNN, 2) LSTM
(Fig. 7. The results of checking an independent dataset of twenty files by trained neural networks: (Classes - the class of the file «according to» the neural network, Proba - the percentage of the contents of the file to the class (left - 0, right - 1))): 1) 1D CNN, 2) LSTM)

Рекомендательно считается, что нейронная сеть пригодна для эксплуатации, если точность её работы превышает 80% на тестовых данных. Однако требования к точности нейронной сети могут зависеть от конкретного вида работы, которую она должна выполнять, от внутренней политики безопасности организации и т.п. Результаты работы нейронной сети можно улучшить, если подготовить для обучения нейронной сети датасет большего объёма и с максимально разнообразным содержимым. Также можно включить в датасет не только ключевые слова, которые могли бы указывать на присутствие

конфиденциальной информации в данных (например, «паспорт», «серия», «номер»), но и список распространённых имён, фамилий и отчеств, а также совместить их с уже имеющимся датасетом для увеличения насыщенности текста. После тестирования нейронная сеть и модуль системы мониторинга безопасности информационной сети могут быть введены в эксплуатацию.

Заключение

Результатом разработки является модуль DLP-системы, позволяющий осуществлять мониторинг трафика корпоративной сети и контролировать пересылку конфиденциальных данных по этой сети. Контроль пересылки осуществляется модулем с нейронной сетью. Веб-интерфейс на Flask-сервере позволяет производить мониторинг инцидентов внутри корпоративной сети.

СПИСОК ЛИТЕРАТУРЫ:

1. Searchinform information security. URL: https://searchinform.ru/informatsionnaya-bezopasnost/dlp-sistemy/ (дата обращения: 20.10.2019).
2. Reviews for Enterprise Data Loss Prevention. URL: https://www.gartner.com/reviews/market/enterprise-data-loss-prevention/vendors (дата обращения: 20.10.2019).
3. Обзор Symantec Data Loss Prevention 12.5. URL: https://www.anti-malware.ru/reviews/Symantec_Data_Loss_Prevention_12_5 (дата обращения: 20.10.2019).
4. Symantec DLP products. URL: https://www.symantec.com/products/dlp (дата обращения: 20.10.2019).
5. Symantec Data Loss Prevention. URL: https://vido.com.ua/article/2530/symantec-data-loss-prevention/ (дата обращения: 20.10.2019).
6. McAfee Total Protection for Data Loss Prevention. URL: http://www.azone-it.ru/mcafee-dlp (дата обращения: 20.10.2019).
7. McAfee Total Protection for DLP. URL: https://www.mcafee.com/enterprise/ru-ru/products/total-protection-for-data-loss-prevention.html (дата обращения: 20.10.2019).
8. Forcepoint DLP products. URL: https://www.forcepoint.com/product/dlp-data-loss-prevention (дата обращения: 20.10.2019).
9. Introduction to Forcepoint DLP Machine Learning. URL: https://www.websense.com/content/support/library/data/v84/machine_learning/first.aspx (дата обращения: 20.10.2019).
10. InfoWatch Traffic Monitor. URL: https://www.infowatch.ru/products/traffic_monitor (дата обращения: 20.10.2019).
11. Системы защиты от утечек конфиденциальной информации (DLP). URL: https://www.anti-malware.ru/security/data-leak-protection (дата обращения: 20.10.2019).
12. Машинное обучение для людей. URL: https://vas3k.ru/blog/machine_learning/ (дата обращения: 20.10.2019).
13. Report on Text Classification using CNN, RNN & HAN. URL: https://medium.com/jatana/report-on-text-classification-using-cnn-rnn-han-f0e887214d5f (дата обращения: 26.02.2020).
14. Python. URL: https://www.python.org/ (дата обращения: 20.10.2019).
15. Keras: The Python Deep Learning library. URL: https://keras.io/ (дата обращения: 20.10.2019).
16. TensorFlow - an end-to-end open source machine learning platform. URL: https://www.tensorflow.org/ (дата обращения: 20.10.2019).
17. Почтовая кухня #2: SMTP. URL: https://habr.com/ru/post/51772/ (дата обращения: 26.02.2020).
18. Flask против Django: почему Flask может быть лучше. URL: https://python-scripts.com/flask-vs-django (дата обращения: 26.02.2020).
19. A Word2Vec Keras tutorial. URL: https://adventuresinmachinelearning.com/word2vec-keras-tutorial/ (дата обращения: 26.02.2020).
20. Keras. Text Preprocessing. Tokenizer. URL: https://keras.io/preprocessing/text/ (дата обращения: 20.10.2019).

REFERENCES:

[1] Searchinform information security. URL: https://searchinform.ru/informatsionnaya-bezopasnost/dlp-sistemy/ (accessed: 20.10.2019).

[2] Reviews for Enterprise Data Loss Prevention. URL: https://www.gartner.com/reviews/market/enterprise-data-loss-prevention/vendors (accessed: 20.10.2019).
[3] Review Symantec Data Loss Prevention 12.5. URL: https://www.anti-malware.ru/reviews/Symantec_Data_Loss_Prevention_12_5 (accessed: 20.10.2019).
[4] Symantec DLP products. URL: https://www.symantec.com/products/dlp (accessed: 20.10.2019).
[5] Symantec Data Loss Prevention. URL: https://vido.com.ua/article/2530/symantec-data-loss-prevention/ (accessed: 20.10.2019).
[6] McAfee Total Protection for Data Loss Prevention. URL: http://www.azone-it.ru/mcafee-dlp (accessed: 20.10.2019).
[7] McAfee Total Protection for DLP. URL: https://www.mcafee.com/enterprise/ru-ru/products/total-protection-for-data-loss-prevention.html (accessed: 20.10.2019).
[8] Forcepoint DLP products. URL: https://www.forcepoint.com/product/dlp-data-loss-prevention (accessed: 20.10.2019).
[9] Introduction to Forcepoint DLP Machine Learning. URL: https://www.websense.com/content/support/library/data/v84/machine_learning/first.aspx (accessed: 20.10.2019).
[10] InfoWatch Traffic Monitor. URL: https://www.infowatch.ru/products/traffic_monitor (accessed: 20.10.2019).
[11] Systems of protection against leaks of confidential information (DLP). URL: https://www.anti-malware.ru/security/data-leak-protection (accessed: 20.10.2019).
[12] Machine learning for people. URL: https://vas3k.ru/blog/machine_learning/ (accessed: 20.10.2019).
[13] Report on Text Classification using CNN, RNN & HAN. URL: https://medium.com/jatana/report-on-text-classification-using-cnn-rnn-han-f0e887214d5f (accessed: 26.02.2020).
[14] Python. URL: https://www.python.org/ (accessed: 20.10.2019).
[15] Keras: The Python Deep Learning library. URL: https://keras.io/ (accessed: 20.10.2019).
[16] TensorFlow - an end-to-end open source machine learning platform. URL: https://www.tensorflow.org/ (accessed: 20.10.2019).
[17] Mail kitchen #2: SMTP. URL: https://habr.com/ru/post/51772/ (accessed: 26.02.2020).
[18] Flask vs Django: why Flask could be better. URL: https://python-scripts.com/flask-vs-django (accessed: 26.02.2020).
[19] A Word2Vec Keras tutorial. URL: https://adventuresinmachinelearning.com/word2vec-keras-tutorial/ (accessed: 26.02.2020).
[20] Keras. Text Preprocessing. Tokenizer. URL: https://keras.io/preprocessing/text/ (accessed: 20.10.2019).

Поступила в редакцию – 13 декабря 2019 г. Окончательный вариант – 29 февраля 2020 г.
Received – December 13, 2019. The final version – February 29, 2020.