Инженерный вестник Дона, №1 (2023)
ivdon.ru/ru/magazine/archive/n1y2023/8163

Подход к работе системы защиты сети передачи данных от компьютерных атак на основе гибридной нейронной сети

В.Х. Федоров[1], Д.Ю. Васюков[2], О.С Лаута[3], Е.Г. Баленко[1], Иванов Д.А.[2]

[1]Донской государственный аграрный университет, пос. Персиановский, Ростовская область
[2]Военная орденов Жукова и Ленина Краснознаменная академия связи им. С.М. Буденного, Санкт-Петербург
[3]Государственный университет морского и речного флота имени адмирала С.О. Макарова, Санкт-Петербург

Аннотация: В статье описан подход к работе системы защиты сети передачи данных от компьютерных атак (КА) на основе гибридной нейронной сети. В качестве метода машинного обучения предлагается использовать гибридную нейронную сеть. Для вычисления выходного значения сигналов нейронной сети, используется функция активации. Модель нейронной сети состоит из рекуррентных ячеек с долгой краткосрочной памятью. Эксперименты продемонстрировали, что предлагаемая система защиты сети при обнаружении компьютерных атак на основе оценки самоподобия параметров функционирования системы с использованием фрактальных показателей и прогнозирования факта воздействия кибератак путем применения предложенной структуры нейронной сети LSTM обладает достаточно высокой эффективностью при обнаружении как известных, так и неизвестных КА. Вероятность обнаружения известных КА равна 0,96, а атаки “нулевого дня” – 0,8.
Ключевые слова: сеть передачи данных, компьютерная атака, нейронная сеть, система защиты, сетевой трафик, автокодировщик, точность, полнота, обнаружение, классификатор, самоподобие, рекуррентные ячейки с долгой краткосрочной памятью.

Использование в сети передачи данных (СПД) информационных и коммуникационных технологий для сбора информации способствует возможности третьего лица воздействовать на сети путем реализации компьютерных атак (КА).

Воздействия КА приводит к появлению в СПД девиантной активности трафика, для постоянного обнаружения и мониторинга в СПД необходимо учитывать наличие большого количества сетевых маршрутов. Все это послужило поводом для разработки методики раннего обнаружения компьютерных атак в сетевом трафике СПД (рис. 1). В качестве метода машинного обучения предлагается использовать гибридную нейронную сеть (рис. 2 и 3).

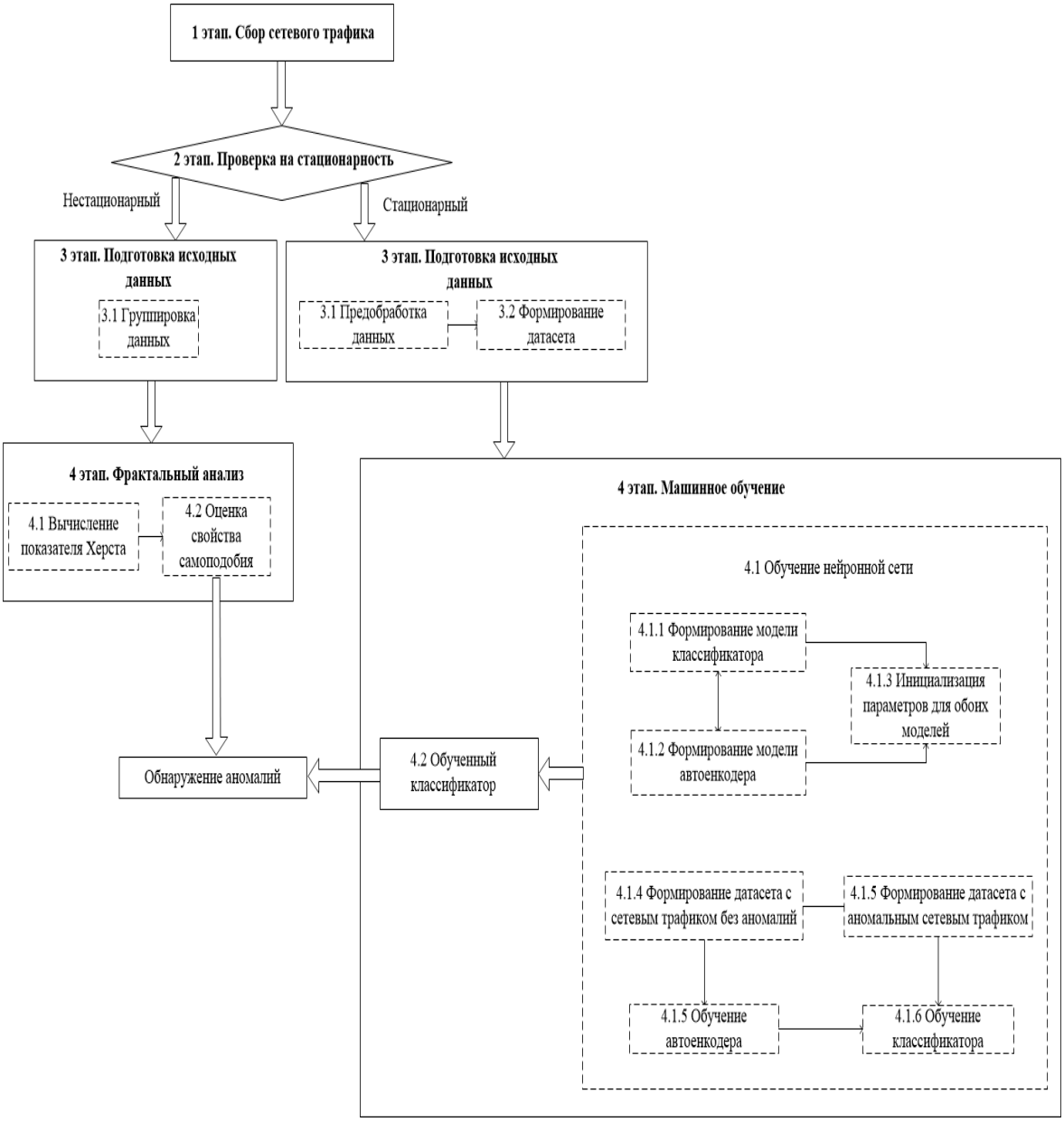

Рис.1. Структура методики раннего обнаружения компьютерных атак в сетевом трафике СПД

Для вычисления выходного значения сигналов нейронной сети, используется функция активации. Модель нейронной сети состоит из рекуррентных ячеек с долгой краткосрочной памятью – Long Short-Term Memory (LSTM) и GRU [1, 2]. На вход нейронной сети подаются данные размерностью до 699 символов. Выходных слоев у нейронной сети несколько. Выходной слой у автокодировщика имеет точно такую же

размерность, как и входной. Выходной слой у классификатора один. Он определяет, является ли запрос аномальным или легитимным.

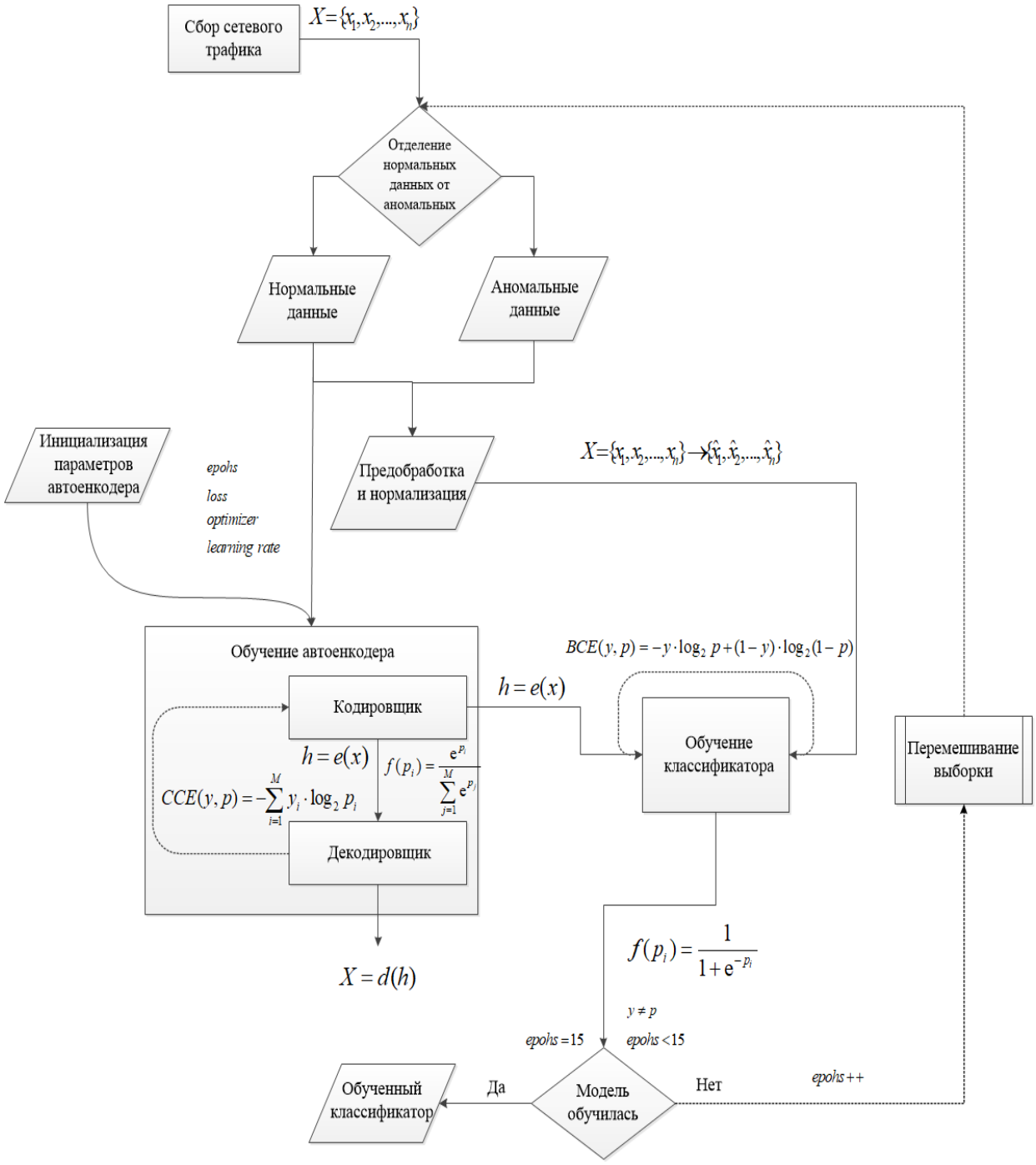

Рис.2. Гибридная нейронная сеть

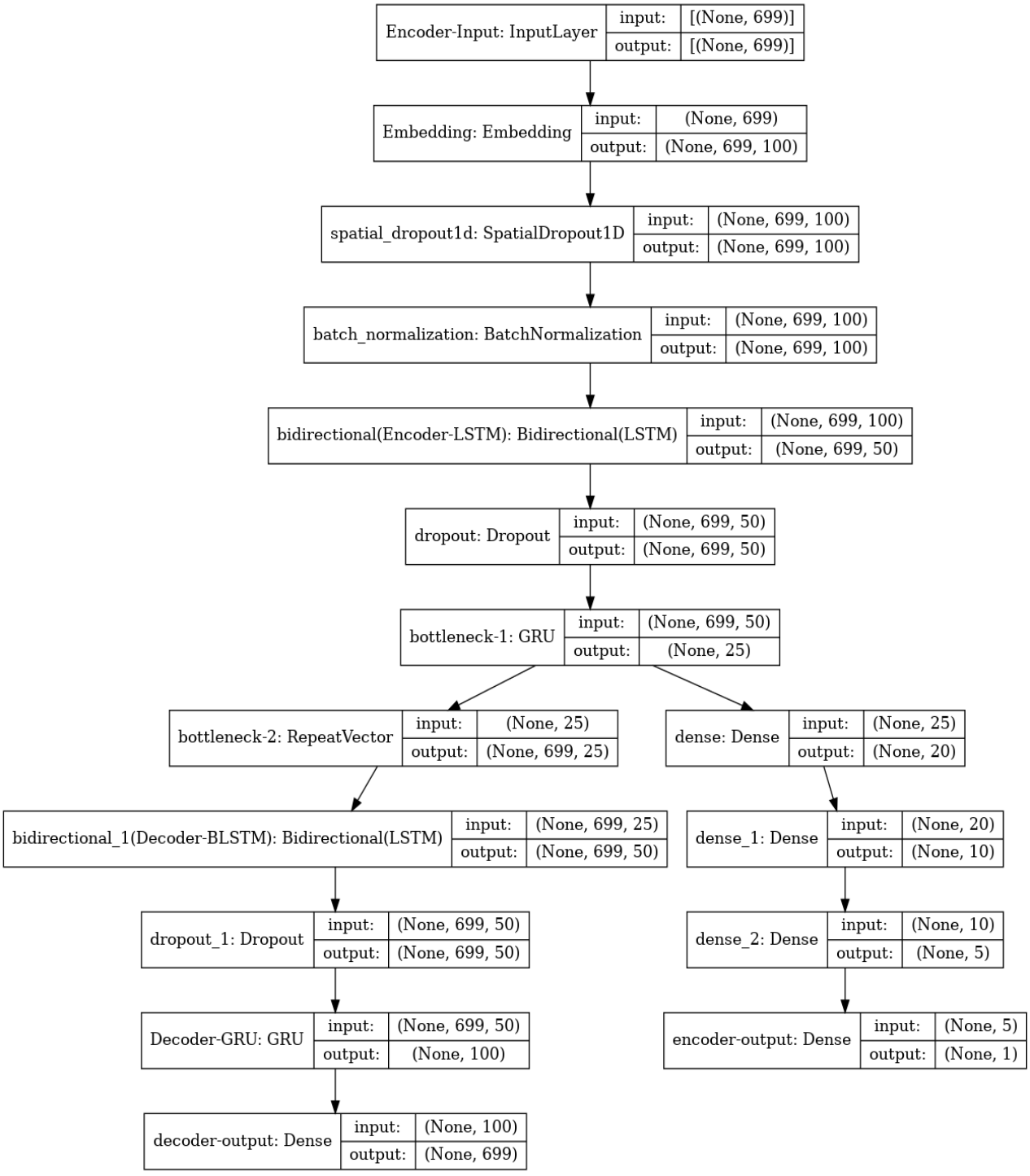

Рис.3. Граф гибридной нейронной сети

В качестве слоев автокодировщика, используются рекуррентные ячейки с долгой краткосрочной памятью – LSTM и GRU.

Свойство рекуррентности позволяет искусственной нейронной сети

«обращаться» к результатам своей работы в прошлом, делать анализ предикций. Тем самым контекст решений в будущем будет зависеть не только от первичного глубокого обучения нейронной сети долгой краткосрочной памяти (LSTM), но и её дальнейшей работы в потоке [3].

Нейронные сети долгой краткосрочной памяти являются подтипом более общих рекуррентных нейронных сетей (РНС). Основной особенностью повторяющихся нейронных сетей является их способность сохранять состояние ячейки или информацию, для дальнейшего использования в сети, которые делают их подходящими для анализа временных данных, изменяемых с течением времени.

Сеть долгой краткосрочной памяти может удалять информацию из состояния ячейки; называемыми gates (фильтрами), которые регулируют структуры. Фильтры состоят из нескольких условий, позволявших пропускать информацию. Они состоят из слоя сигмоидальной нейронной сети и операции поточечного умножения. Сигмоидальный слой возвращает числа от нуля до единицы, которые обозначают, какую долю каждого блока информации следует пропустить дальше по сети. Ноль в данном случае означает: «не пропускать ничего», единица – «пропустить все» [4].

Программная реализация разработанной методики разработана на языке программирования Python с использованием библиотеки Pandas, с помощью которой осуществлялись обработки и анализа данных. Библиотека Pandas написана на языках программирования Си, Cython, и Python, которые делают Python приоритетным инструментом для оценки данных и дают возможность на высококачественном показателе строить сводные таблицы, выполнять группировки, предоставлять удобный доступ к табличным данным.

Кроме того, помимо библиотеки Pandas использовалась библиотека NumPy, которая представляет собой инструмент более низкого уровня,

обеспечивающий работу с высокоуровневыми математическими функциями, а также с многомерными массивами (тензорами). Графики строились с помощью модуля научной графика в Python (Matplotlib) на основе полученного набора данных. Расчеты производились в интегрированной среде разработки Jupiter notebook [5, 6].

На рисунке 4 изображена блок-схема, отражающая этапы формирования данных и обучения гибридной модели нейронной сети.

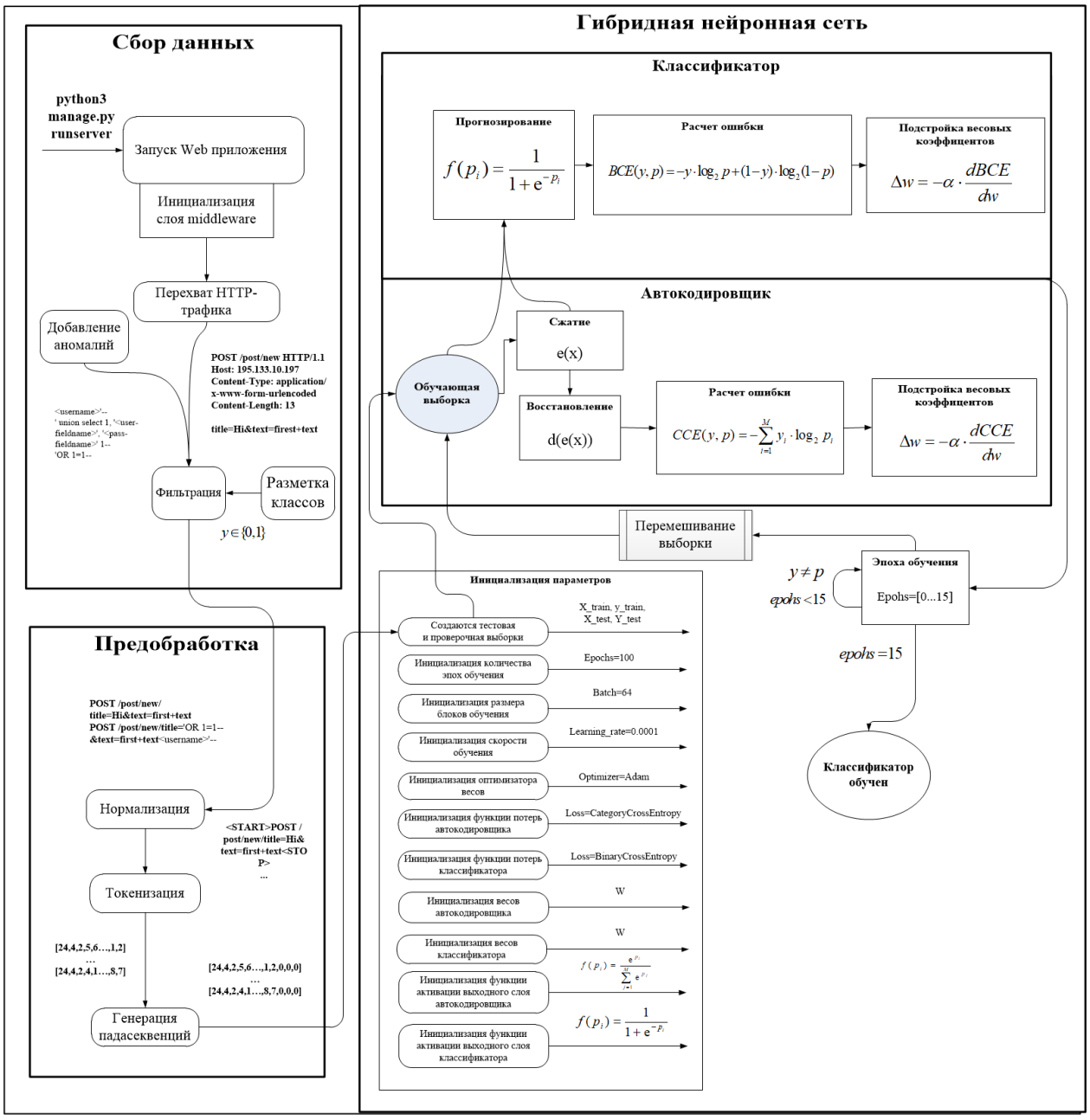

Рис. 4. Блок-схема этапов обучения гибридной нейронной сети

Для того, чтобы сформировать датасет данных, предназначенных для обучения нейронной сети, на языке программирования python реализовано web-приложение, способное перехватывать любые пользовательские запросы с помощью промежуточного слоя middleware. Такой подход позволяет обрабатывать запросы из браузера, прежде чем они достигнут представления Django (сервера), а также ответы от представлений до того, как они возвращаются в браузер.

Следующий этап обучения гибридной нейронной сети включает в себя нормализацию данных. Запросы оборачиваются специальными токенами <START> и <STOP>, что задаёт верное вероятностное распределение над последовательностями разной длины [7]. Для проверки стационарности HTTP-трафика, проведен эксперимент, который заключался в построении графика распределения длин между двумя одинаковыми символами (рис. 5) и оценки стационарности получившегося ряда с помощью теста Дики-Фуллера.

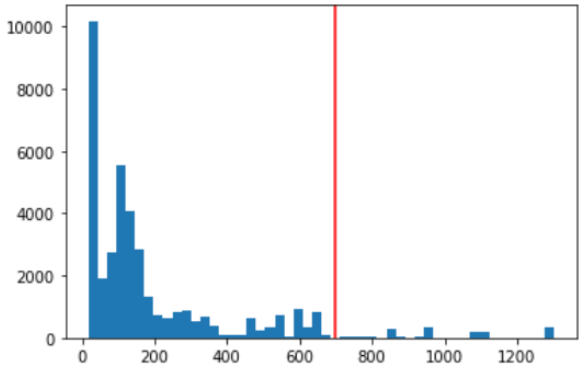

Рис. 5. Распределение длин легитимных запросов

Далее произведена предобработка и нормализация получившейся выборки. Поскольку протокол HTTP – текстовый протокол, использовалось векторное представление символов [8, 9]. Для этого сперва осуществляется замена символов, встречающихся в датасете на числовой эквивалент, который не имеет самостоятельного смысла/значения для внешнего или внутреннего использования (токенизировать), а затем переводятся слова в последовательность секвенций (рис. 6), с помощью токенизации.

```
data_train = pad_sequences(X_train_sequences, maxlen=max_len_str, padding='post')
data_test = pad_sequences(X_test_sequences, maxlen=max_len_str, padding='post')
```

```
data_train
```

```
array([[24,  4,  2, ...,  0,  0,  0],
       [24,  4,  2, ...,  0,  0,  0],
       [24,  4,  2, ...,  0,  0,  0],
       ...,
       [24,  4,  2, ...,  0,  0,  0],
       [24,  4,  2, ...,  0,  0,  0],
       [24,  4,  2, ...,  0,  0,  0]], dtype=int32)
```

Рис.6. Секвенции

При этом, основополагающим обстоятельством является то, что все секвенции одной длиной, если запрос меньше длины секвенции, то заполняются нулями оставшиеся символы.

Далее секвенции подаются на вход гибридной нейронной сети, и подбираются гиперпараметры. Но перед этим необходимо настроить среду выполнения таким образом, чтобы все вычисления происходили на GPU. Для этой цели достаточно установить библиотеки, строго определенных версий:

conda create --name tf python=3.8

conda activate tf

conda install cudatoolkit=10.0.130

conda install cudnn=7.6.0=cuda10.0_0

pip install --upgrade tensorflow-gpu

sudo apt install libcudnn8

Датасет, предназначенный для обучения нейронной сети, должен включать как легитимный трафик, так и аномальный. На вход автокодировщика подается только легитимный трафик. На вход классификатора подается легитимный, аномальный трафик и скрытые латентные представления, полученные с автокодировщика после кодирования информации (рис. 7).

```
Trial 27 Complete [00h 27m 58s]
decoder-output_loss: 19800.93359375

Best decoder-output_loss So Far: 19800.93359375
Total elapsed time: 05h 00m 36s

Search: Running Trial #28

Hyperparameter    |Value             |Best Value So Far
decoder-output    |0.0014            |0.0039
encoder-output    |90                |45
learning_rate     |1e-06             |1e-05
tuner/epochs      |10                |10
tuner/initial_e...|0                 |0
tuner/bracket     |0                 |0
tuner/round       |0                 |0

Epoch 1/10
1351/1351 [==============================] - 218s 156ms/step - loss: 92.7079 - encoder-
Epoch 2/10
1351/1351 [==============================] - 208s 154ms/step - loss: 92.4347 - encoder-
Epoch 3/10
1026/1351 [=====================>........] - ETA: 43s - loss: 92.1161 - encoder-output_
```

Рис. 7. Подбор гиперпараметров нейронной сети

Подбор параметров осуществляется таким образом, чтобы функция потерь при обучении автокодировщика уменьшалась, при этом точность классификатора росла.

На рисунке 8 продемонстрирован рост точности и снижение потерь на 30 эпохах обучения.

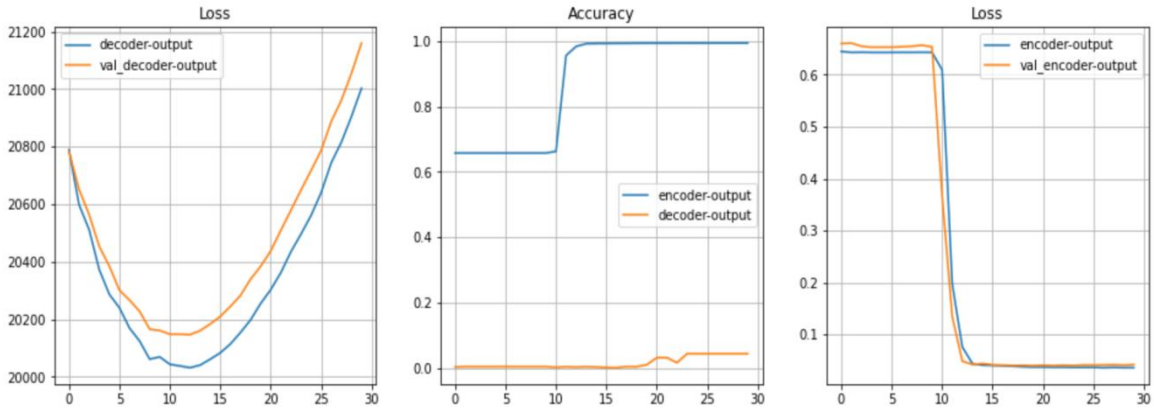

Рис. 8. Обучение декодера и классификатора на 30 эпохах

После обучения нейронной сети, проведен эксперимент по оценке точности и полноты. Сперва использовался датасет с КА такого же типа, как в датасете при обучении модели. Результат обнаружения аномалий – 96,9% (рис. 9).

```
scores = encoder.evaluate(X_pad, data_last['anomaly'].values, verbose=1)
print('Точность: {}% \nLoss: {}'.format(scores[1]*100, 1 - scores[1]))
```

```
1799/1799 [==============================] - 52s 29ms/step - loss: 0.0478 - binary_accuracy: 0.9850
Точность: 96.90268635749817%
Loss: 0.03097313642501831
```

Рис.9. Оценка точности алгоритма на известных аномалиях

Затем был сформирован новый датасет, с компьютерными атаками раннее неизвестными классификатору (атаки нулевого дня) и алгоритм распознал 99% неизвестных раннее атак (атаки 0-го дня), и верно определил, что 99% легитимных запросов не являются аномальными (рис. 10).

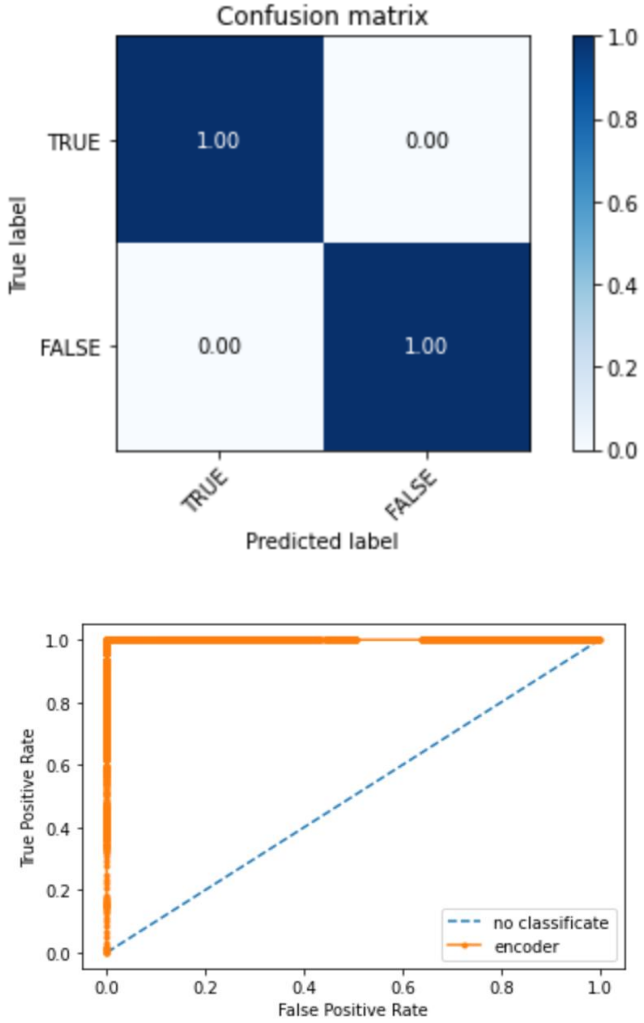

Рис. 10. Оценка точности алгоритма на неизвестных аномалиях

Система допускает ложные срабатывания. В данном случае, 10 запросов были отброшены нейронной сетью (рис. 11).

```
data_last[(data_last['pred']==True) & (data_last['anomaly']==False)]
```

	data	anomaly	pred
661	<START>POST /post/new/title=Действительно,+бол...	0	True
1153	<START>POST /post/new/title=англоязычный+форум...	0	True
2191	<START>POST /post/new/title=Ведь+в+понятиетext...	0	True
2662	<START>POST /post/new/title=ладно,как+начну+сн...	0	True
5775	<START>POST /post/new/title=The+cyber-physical...	0	True
6800	<START>POST /post/new/title=PopMech+and+its+an...	0	True
11024	<START>POST /post/new/title=ладно,как+начну+сн...	0	True
22111	<START>POST /post/new/title=Google+Search+Rank...	0	True
27957	<START>POST /post/new/title=огромное+спасибо+з...	0	True
34286	<START>POST /post/new/title=BlueWillowPlate++t...	0	True

Рис. 11. Пример ложных срабатываний

Вышеописанная система сравнительной оценки эффективности была проведена на основе его сравнения с другими известными системами, например, IDS и IPS, которые при обнаружений КА (аномалий) используют сигнатурный метод, статистический метод и методы машинного обучения. Сигнатурные методы используют заранее составленные требования, которые имеют высокую точность обнаружения известных типов кибернетических атак. Однако они не умеют выявлять новые, неизвестные типы атак, включая advanced persistent threat (таргетированные атаки). Помимо этого, они имеют средние показатели по отсутствию ложного обнаружения [10].

Статистические методы уступают сигнатурным методам по скорости обнаружения и точности обнаружения известных атак, используя накопленную статистику, хотя в ряде случаев они способны обнаруживать неизвестные атаки. По ложному обнаружению они имеют примерно такие же возможности, как и сигнатурные методы.

В настоящее время методы машинного обучения являются разнообразными и хорошо развитыми. Несмотря на то, что процесс обнаружения атак в этих методах, обязательно предшествует процессу обучения на контрольной выборке, анализ показывает, что по скорости обнаружения атак эти методы уступают сигнатурным методам. Однако эти методы имеют хорошую точность обнаружения неизвестных атак и более высокую точность обнаружения известных атак. Несмотря на это, доля ложных срабатываний в методах машинного обучения является крайне низкой.

Эксперименты продемонстрировали, что предлагаемая система защиты СПД при обнаружении КА на основе оценки самоподобия параметров функционирования системы с использованием фрактальных показателей и прогнозирования факта воздействия кибератак путем применения предложенной структуры нейронной сети LSTM обладает достаточно высокой эффективностью при обнаружении как известных, так и неизвестных КА. Вероятность обнаружения известных КА равна 0,96, а атаки “нулевого дня” - 0,8.

Литература

1. Курило А.А., Сорокин М.А., Стародубцев Ю.И. Модель системы связи как источника отличительных признаков // Инженерный вестник Дона, 2021. № 3. URL: ivdon.ru/ru/magazine/archive/n3y2021/6880.

2. Курило А.А., Сорокин М.А., Стародубцев Ю.И. Методика обработки результатов мониторинга с динамически изменяемым уровнем разрешающей способности базы данных // Инженерный вестник Дона, 2021, № 3. URL: ivdon.ru/ru/magazine/archive/n3y2021/6882.

3. Gers F.A., Schmidhuber J., Cummins F. Learning to Forget: Continual Prediction with LSTM // Neural Computation, 2000. Vol. 12, №. 10. pp. 2451-2471.

4. Graves A., Schmidhuber J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures // Neural Networks, 2005. Vol. 18, № 5–6. pp. 602–610.

5. Nurul A.H., Zaheera Z.A. Risk Assessment Method for Insider Threats in Cyber Security: A Review // International Journal of Advanced Computer Science and Applications, 2018, Vol. 9, № 11.

6. Tankard C. Advanced persistent threats and how to monitor and deter them // Network Security, 2011, № 8, pp. 16–19.

7. Kotenko, I.; Lauta, O.; Kribel, K.; Saenko, I. LSTM neural networks for detecting anomalies caused by web application cyber attacks //Front. Artif. Intell. Appl., 2021, 337, pp. 127–140.

8. Савищенко Н.В., Остроумов О.А. Расчет оптимального и рационального числа ветвей разнесения в каналах связи с аддитивным белым гауссовским шумом и общими замирания и Райса-Накагами // Информационно-управляющие системы, 2015, № 6. с. 71-80. doi:10.15217/issn1684-8853.2015.6.71.

9. Sinjuk A. D., Ostroumov O. A. Theorem about key capacity of a communication network. Informatsionno-upravliaiushchie sistemy [Information and Control Systems], 2018, № 5, pp. 79–87. doi:10.31799/1684-8853-2018-5-79-87.

10. Котенко И.В., Саенко И.Б., Лаута О.С., Крибель А.М. Методика обнаружения аномалий и кибератак на основе интеграции методов фрактального анализа и машинного обучения // Информатика и автоматизация, 2022, Т. 21, № 6. с. 1328-1358.

References

1. Kurilo A.A., Sorokin M.A., Starodubcev Yu.I. Inzhenernyj vestnik Dona, 2021, № 3. URL: ivdon.ru/ru/magazine/archive/n3y2021/6880.

2. Kurilo A.A., Sorokin M.A., Starodubcev Yu.I. Inzhenernyj vestnik Dona, 2021, № 3. URL: ivdon.ru/ru/magazine/archive/n3y2021/6882.

3. Gers F.A., Schmidhuber J., Cummins F. Learning to Forget: Continual Prediction with LSTM. Neural Computation. 2000. Vol. 12, №. 10. pp. 2451-2471.

4. Graves A., Schmidhuber J. Neural Networks. 2005. Vol. 18, № 5–6. pp. 602–610.

5. Nurul A.H., Zaheera Z.A. International Journal of Advanced Computer Science and Applications, 2018, Vol. 9, № 11.

6. Tankard C. Network Security, 2011, № 8. pp. 16–19.

7. Kotenko, I.; Lauta, O.; Kribel, K.; Saenko, I. Front. Artif. Intell. Appl. 2021, 337. pp. 127–140.

8. Savishhenko N.V., Ostroumov O.A. Informatsionno-upravliaiushchie sistemy, 2015, № 6. c. 71-80. doi: 10.15217/issn1684-8853.2015.6.71.

9. Sinyuk A.D., Ostroumov O.A. Informatsionno-upravliaiushchie sistemy, 2018, № 5, pp. 79–87. doi:10.31799/1684-8853-2018-5-79-87.

10. Kotenko I.V., Saenko I.B., Lauta O.S., Kribel` A.M. Informatika i avtomatizaciya, 2022, T. 21, № 6. pp. 1328-1358.