|  |
| --- |
| Московский государственный университет имени М.В. Ломоносова  *Факультет вычислительной математики и кибернетики* |
|  |
| Яркова Юлия Сергеевна  **Применение рекуррентных сверточных сетей для анализа аномалий во временных рядах***Отчет по преддипломной практике*  студента образовательной программы «Прикладная математика и информатика»  по направлению подготовки *01.03.02 Прикладная математика и информатик*   |  |  | | --- | --- | | Москва 2023 год | Руководитель  Доцент, кандидат физико-математических наук  Березин С.Б. | |

.

1. Постановка задачи и характеристика данных

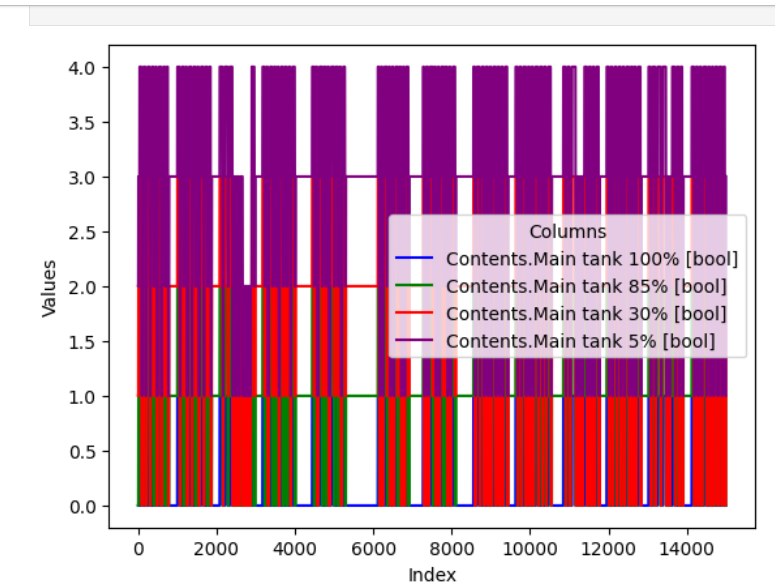
Имеется 4 бинарных датчика уровня, которые являются датчиками уровня питательного раствора в основном баке, который используется для полива растений, в количестве 100%, 85%, 30% и 5%. Когда идет полив, жикость постепенно убывает из бака, а когда полив заканчивается, жидкость постепенно возвращается обратно. Датчик включен, когда количество питательного раствора в основном баке соответствует не меньше процента, указанного выше. Аномалии могут быть связаны с постепенным засорением фильтров, в результате чего возврат раствора в бак замедляется и интервалы, в которые датчик отключен, постепенно удлиняются. Пример визуализации части сигналов приведен на рисунке ниже: 

Рисунок 1 − Пример визуализации части данных. По горизонтальной оси отложен индекс момента времени, в который произведено измерение, начиная 0. По вертикали отложены состыкованные значения бинарных датчиков.

Цель: применить метод MSCRED для выявления аномалий в работе датчиков [1]..

Моя работа состояла из нескольких этапов:

1. Изучение теоретических основ метода MSCRED и его требований к входным данным;
2. Подготовка данных, полученных с установки, для использования в методе MSCRED;
3. Применение метода MSCRED и анализ полученных результатов.
4. Предобработка данных

Метод MSCRED работает с данными, измеренными через равные промежутки времени. Наши же данные являются неравномерными: имеются как малые промежутки, так и очень большие. Поэтому необходимо было произвести преобразование данных из неравномерных в равномерные с помощью интерполяции. Датчики выдают бинарные показания, поэтому я производила не линейную интерполяцию, а приведение к ближнему, т.е. приравнивала значения показаний датчиков к ближнему времени из исходных данных. Ближнее измерение мы брали из отрезка:

[t\_i - δ/2; t\_i + δ/2], где t\_i – это очередной момент времени, из которых мы составляем равномерные данные. Если на данном отрезке не оказывалось данных, то в показания датчиков записывались -1. Если на отрезке оказалось несколько точек данных, то берется ближайшая точка.

Для определения δ строилась гистограмма (иллюстрации приведены ниже) промежутков между ближайшими измерениями на всем датасете. На вертикальной оси отложено количество временных промежутков размера, приведенного на горизонтальной оси. Был взят наименьший из самых распространенных промежутков, т.е. 30 секунд.

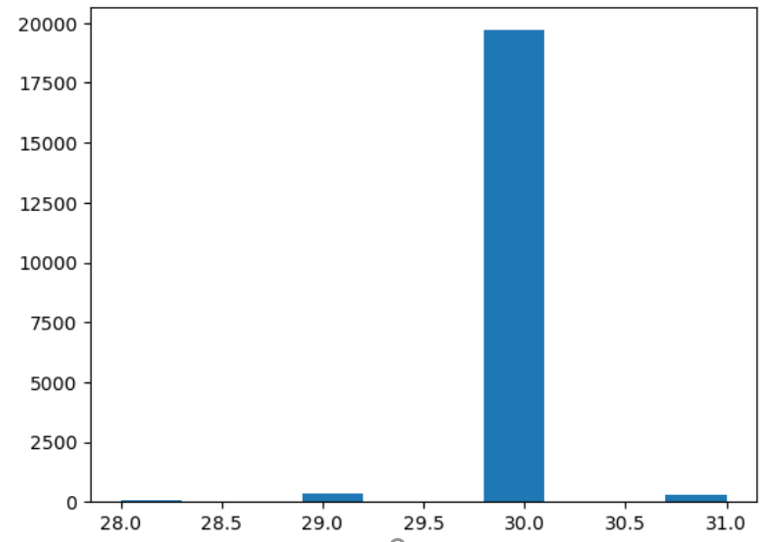


Рисунок 1 − Гистограмма промежутков между ближайшими измерениями (часть 1). На оси абсцисс отмечены размеры промежутков между ближайшими измерениями (в секундах). На оси ординат отмечено количество промежутков, имеющих определенную длину.

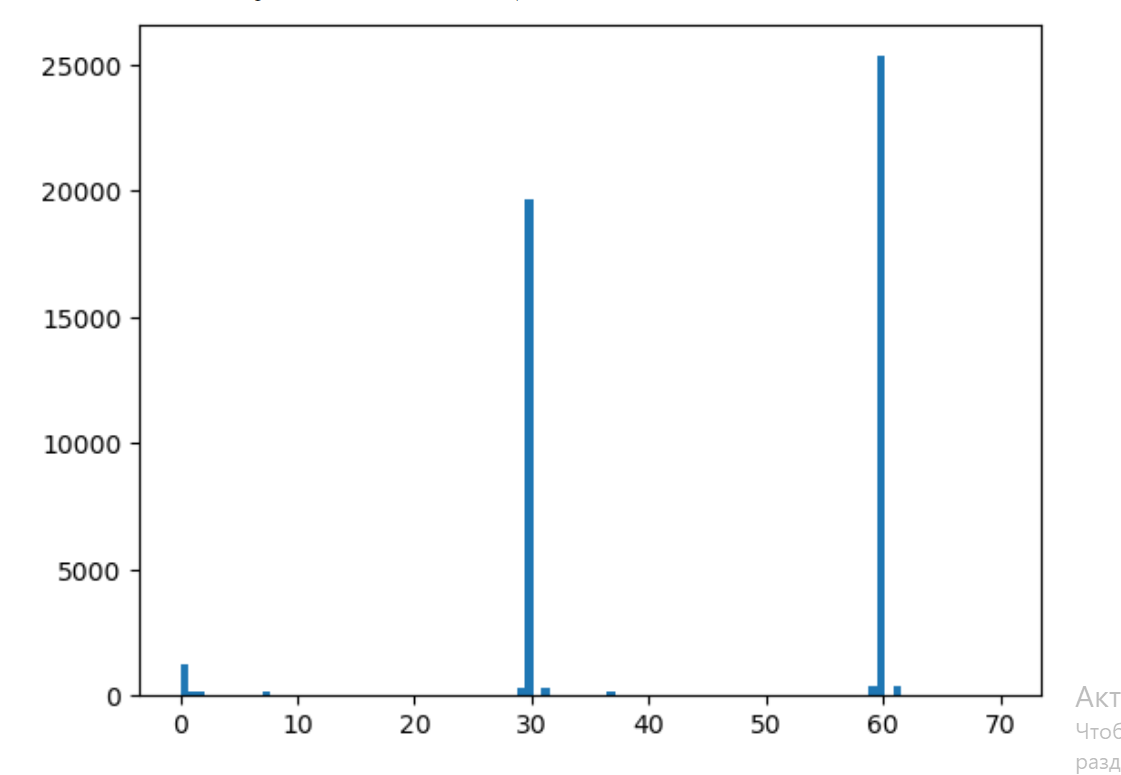


Рисунок 2 − Гистограмма промежутков между ближайшими измерениями (часть 2). На оси абсцисс отмечены размеры промежутков между ближайшими измерениями (в секундах). На оси ординат отмечено количество промежутков, имеющих определенную длину.

При обработке данных выяснилось, что программа, которая использовалась при получении датасета, не гарантирует того, что данные будут расположены в порядке возрастания времени, поэтому была проведена сортировка.

Также в процессе написания преобразователя данных оказалось, что происходит переполнение памяти и сложности с обработкой данных, если записывать преобразованные данные сразу в датафрейм, где датафрейм — это структура данных из пакета pandas [2]. Поэтому данные при преобразовании в равномерные помещались в списки, и только после полного их преобразования на их основе создавался датафрейм.

Скриншот датафрейма с полученными данными приведен ниже:

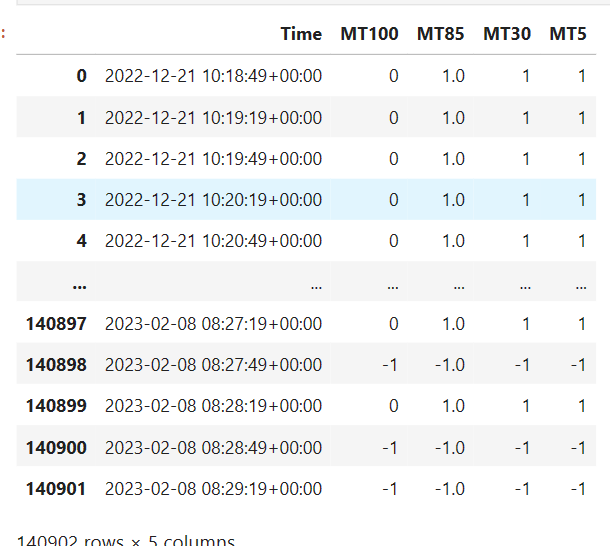


Рисунок 3 − Датафрейм полученных после приведения данных к равномерному виду, заполнения пропусков в данных. Столбцы MT100, MT85, MT30, MT5 − это двоичный сигнал о наличии 100%, 85%, 30%, 5% питательного раствора в основном баке соответственно.

Для анализа данных с помощью метода MSCRED были взяты данные длиной в месяц и шагом в 1 час. Также для использования модели из данных был удален столбец со временем, оставлены только столбцы с показаниями датчиков. В финале датафрейм содержащие данные датчиков был транспонирован. Полученные данные приведены на рисунке ниже:

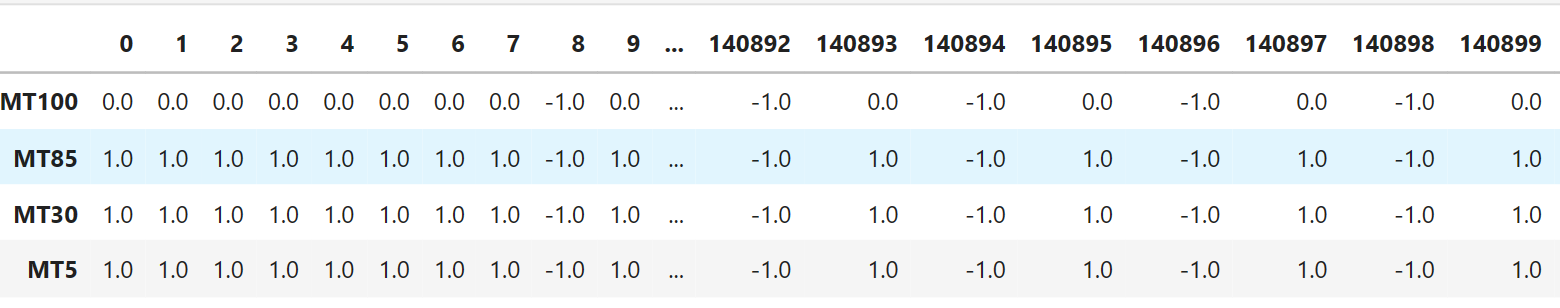


Рисунок 4 − иллюстрация итоговых данных, к которым применялся метод MSCRED

1. Метод MSCRED

MSCRED расшифровывается как MultiScale Convolution Recurrent Encoder Decoder. Модель представляет из себя комбинацию сверточной нейронной сети (CNN) для извлечения локальных признаков из временных данных, представленной LSTM, которая занимается обработкой последовательности. Вначале сверточные слои CNN выполняют извлечение признаков из входных данных, Затем рекуррентные слои RNN обрабатывают эти признаки последовательно, учитывая контекст и зависимости между элементами данных.

* 1. Построение сигнатурных матриц.

Сигнатурные матрицы для метода получаются с помощью трех размеров окон каждого сегмента: 10, 30, 60. Размер окна каждого сегмента − это размер временного интервала, по которому строятся сигнатурные матрицы.

Формула для построения сигнатурных матриц:

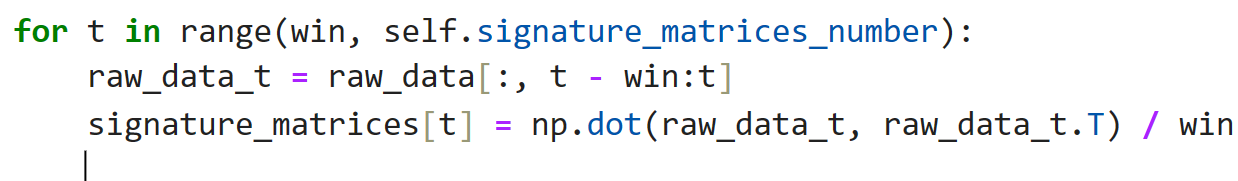
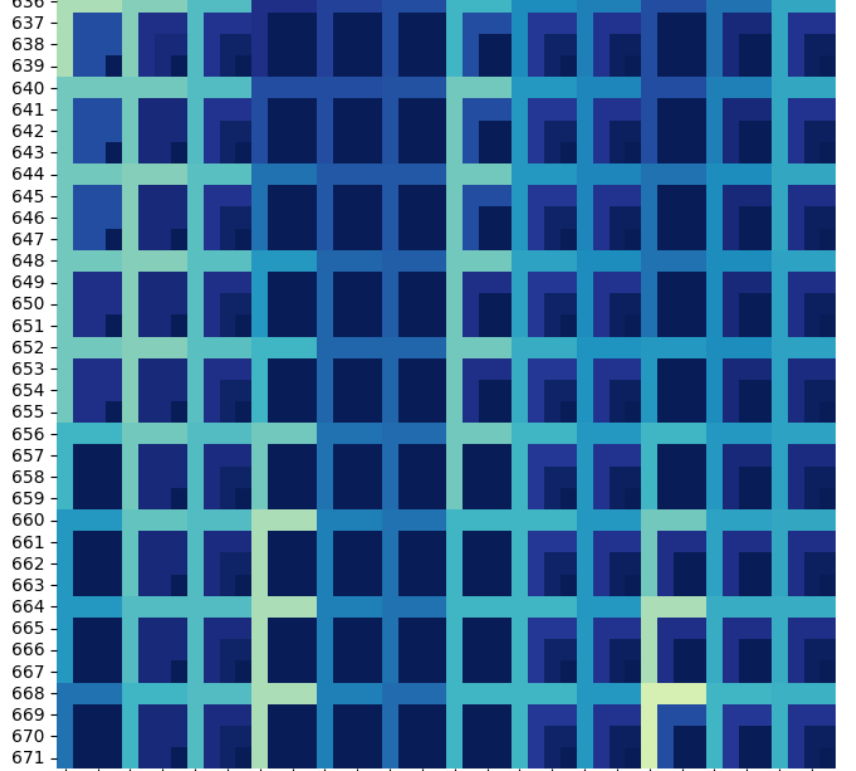


Рисунок 3 − Формула для построения сигнатурных матриц (win − это размер окна, self.signature\_matrices\_number − количество сигнатурных матриц, raw\_data − данные, считанные из полученного датафрейма с данными).

Были построены тепловые карты для полученных 684 сигнатурных матриц. В процессе построения было выяснено, что карты строятся намного быстрее если их выводить путем конкатенации одной матрицы и построения единой тепловой карты, а не построения каждой тепловой карты по отдельности.

Ниже приведен скриншот первых нескольких строк полученных тепловых карт.



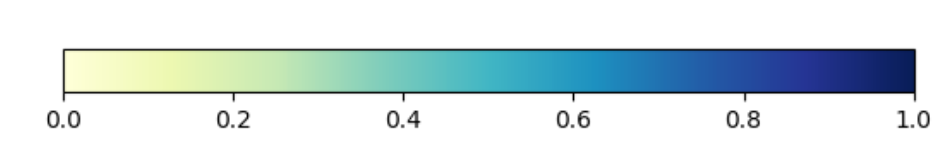


Рисунок 6 − Иллюстрация части тепловых карт сигнатурных матриц. По вертикальной оси отмечен номер сигнатурной матрицы, имеющие единый размер окна и единую неделю получения результатов датчиков. По горизонтальной оси отсчет ведется слева направо, начиная с 0. Матрицы, имеющие номер 3\*j. 3\*j+1, 3\*j+2 − матрицы относящиеся к одной и той же неделе, где j − номер недели, начиная отсчет с 0. Столбцы 3\*j имеют размер окна, равный 10, 3\*j+1 − равный 30, 3\*j+2 − равный 60.

Минимальное значение отмечено светло-желтым цветом и имеет значение 0, максимальное значение отмечено темно-синим цветом и имеет значение 1.

Также для полученных сигнатурных матриц были построены MaxPooling, максимум берется по каждой сигнатурной матрице



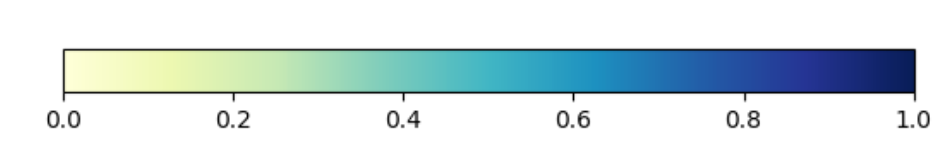
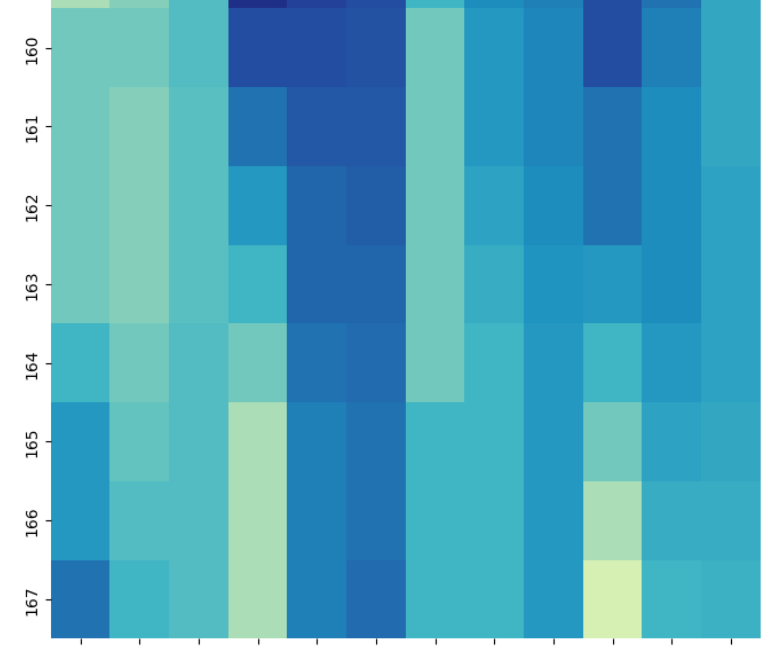


Рисунок 7 − Иллюстрация для построения MaxPooliing для части тепловых матриц. По горизонтальной оси отсчет ведется слева направо, начиная с 0. Матрицы, имеющие номер 3\*j. 3\*j+1, 3\*j+2 − матрицы относящиеся к одной и той же неделе, где j − номер недели, начиная отсчет с 0. Столбцы 3\*j имеют размер окна, равный 10, 3\*j+1 − равный 30, 3\*j+2 − равный 60.

Также были построены MinPooling, где минимум берется по каждой сигнатурной матрице:



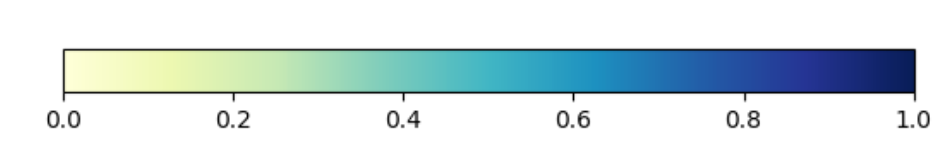


Рисунок 8 − Иллюстрация для построения MinPooliing для части тепловых матриц. По горизонтальной оси отсчет ведется слева направо, начиная с 0. Матрицы, имеющие номер 3\*j. 3\*j+1, 3\*j+2 − матрицы относящиеся к одной и той же неделе, где j − номер недели, начиная отсчет с 0. Столбцы 3\*j имеют размер окна, равный 10, 3\*j+1 − равный 30, 3\*j+2 − равный 60.

* 1. Применение нейронной сети

Блок-схема архитектуры нейронной сети приведена на рисунке ниже:

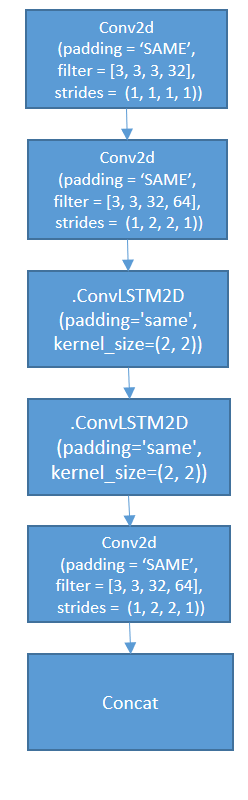


Рисунок 9 − Блок-схема архитектуры нейронной сети

Использована ConvLSTM, содержащая 2 слоя конволюции, 2 слоя LSTM и 1 слой деконволюции.

Использована функция потерь указана на рисунке ниже:

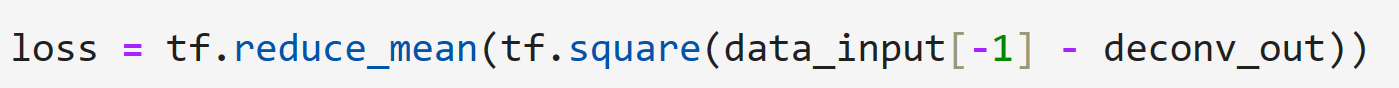


Рисунок 9 − Используемая функция потерь, где deconv\_out − выход слоя деконволюции, data\_input − полученные ранее сигнатурные матрицы, которые поступают на вход нейронной сети.

Значение функции потерь во время обучения приведено на рисунке ниже:

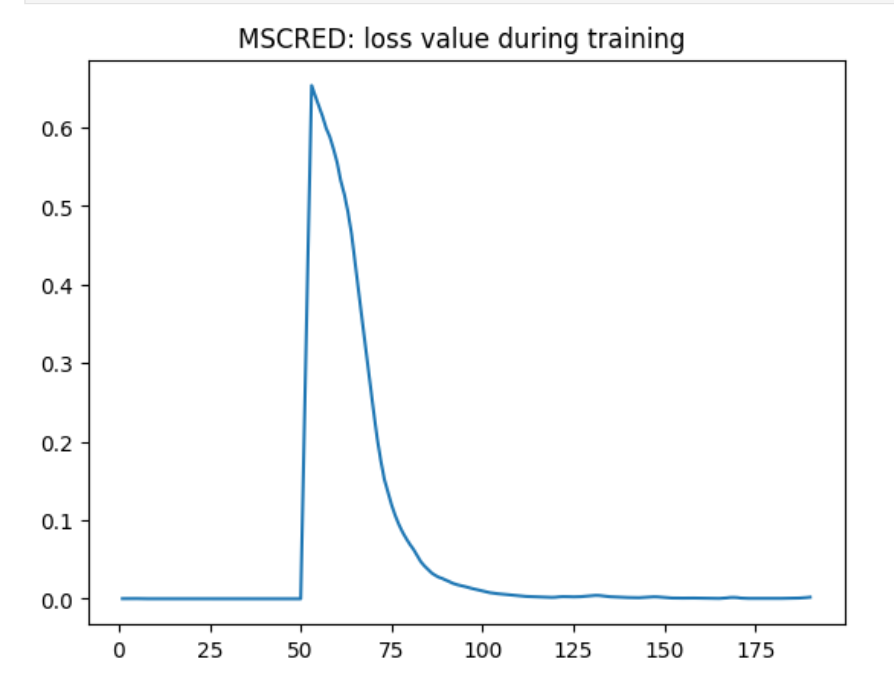


Рисунок 9  значение функции потерь во время этапа обучения модели. По горизонтали отмечен номер тренировочных данных. По вертикали отмечено значение функции потерь.

В качестве оптимизатора используем AdamOptimizer.

Полученная сверточная нейронная сеть успешно обучилась на данных. Ниже приведен график полученных предсказаний:

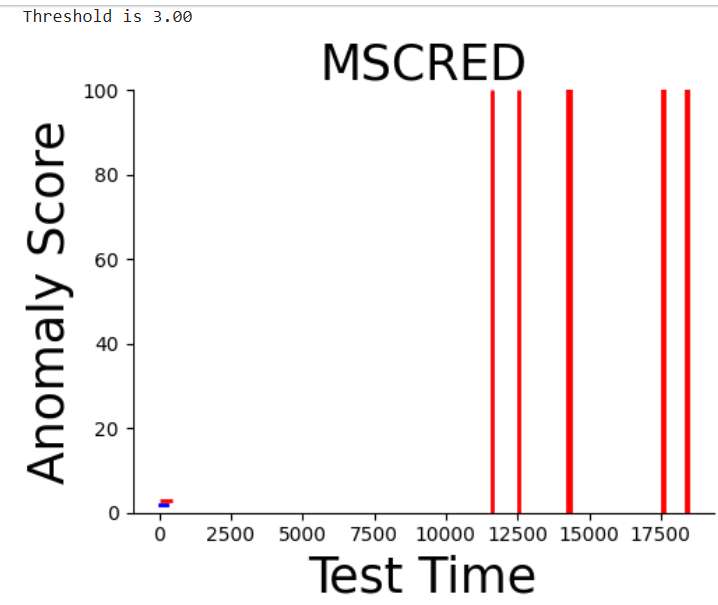


Рисунок 9 − График оценки аномалий. По вертикали отмечены баллы аномалий. По горизонтали номер тестовых данных. Данные, на которых моделью отмечены аномалии имеют балл аномалий выше порогового значения

**Выводы**

В ходе преддипломной практики:

1. Изменены скрипты существующей реализации метода MSCRED под данные;
2. Подготовлены данные;
3. Обновленный скрипт успешно применен к реальным данным, проанализированы полученные результаты;

Библиографический список

1. Chuxu Zhang, Dongjin Song, Yuncong Chen , Xinyang Feng, Cristian Lumezanu , Wei Cheng , Jingchao Ni , Bo Zong , Haifeng Chen , Nitesh V. Chawla A Deep Neural Network for Unsupervised Anomaly Detection and Diagnosis in Multivariate Time Series Data //.
2. pandas.pydata.org URL: https://pandas.pydata.org/ (дата обращения: 22.12.2023).