

Создание LSTM модели для прогнозирования аномалии

Creation of LSTM model for anomaly prediction

Н. О. Костюченко, В. Б. Косогова, С. Р. Бектимиров

Омский государственный технический университет, Омск, Российская Федерация

N. O. Kostyuchenko, V. B. Kosogova, S. R. Bektimirov

Omsk State Technical University, Omsk, Russian Federation

Аннотация. Одним из перспективных направлений в промышленности стала предиктивная аналитика. Она помогает выявлять потенциальные неисправности оборудования и его компонентов, прогнозируя возможные поломки задолго до их возникновения. Раннее обнаружение аномалий позволяет избежать серьезных последствий, а обслуживание агрегатов с выявленными отклонениями обходится значительно дешевле, чем их ремонт или полная замена. Такой подход позволяет оптимизировать бизнес-процессы, повышать эффективность работы предприятий и минимизировать возможные риски. Сейчас же многие компании создают и внедряют системы описательной и предиктивной аналитики. Эти системы анализируют накопленные данные и прогнозируют будущие события, применяя методы математической статистики, моделирования и машинного обучения.

Ключевые слова: предиктивная аналитика, алгоритм, определение отклонений, аномалия, оборудование, предсказание

Abstract. Predictive analytics has become one of the most promising trends in the industry. It helps identify potential malfunctions of equipment and its components, predicting possible breakdowns long before they occur. Early detection of anomalies allows avoiding serious consequences, and maintenance of units with detected abnormalities is much cheaper than their repair or complete replacement. This approach allows optimizing business processes, increasing the efficiency of enterprises and minimizing possible risks. Now many companies create and implement systems of descriptive and predictive analytics. These systems analyze accumulated data and predict future events by applying methods of mathematical statistics, modeling and machine learning.

Keywords: predictive analytics, algorithm, outlier detection, anomaly, hardware, prediction

Введение

В современном мире обработка временных рядов играет ключевую роль в анализе данных, особенно при обнаружении аномалий в сложных системах. Одним из эффективных инструментов для решения этой задачи являются рекуррентные нейронные сети (RNN), а в частности – сети с долгой краткосрочной памятью (LSTM).

Целью данной статьи является создание модели машинного обучения для прогнозирования аномалии на основе LSTM алгоритма.

Для реализации данной темы необходимо выполнить следующие задачи:

- 1) сбор и обработка данных;
- 2) реализация модели автоэнкодера;
- 3) линейный прогноз;
- 4) построение модели LSTM.

Теория

1. Описание автоэнкодера

Многослойные сети-автоэнкодеры – глубокие нейронные сети специальной архитектуры. Многослойный автоэнкодер представляет собой специальный вид многослойной сети прямого распространения – многослойный симметричный персептрон, содержащий несколько внутренних слоев уменьшающегося размера и слой «бутылочная горловина» в середине сети. Этот слой разделяет автоэнкодер на два сегмента: энкодер и декодер [3].

После прохождения через энкодер в узком слое формируется вектор, компоненты которого представляют собой «признаки» – обобщённые характеристики объекта (или код). Эти признаки извлекаются из исходных данных и содержат важную, но избыточную информацию, позволяющую описать объект в пространстве меньшей размерности, называемом скрытым (latent) пространством. Такой процесс достигается за счёт устранения избыточности, а также выявления скрытых взаимосвязей и корреляций в исходных данных.

Декодер выполняет обратное преобразование, восстанавливая данные из скрытого представления, полученного от энкодера. Он также состоит из нескольких слоёв нейронов, но их количество увеличивается на каждом этапе. Декодер обучается реконструировать входные данные, завершая процесс автоэнкодирования, при котором выходные данные стремятся максимально точно соответствовать исходным.

Во время обучения автоэнкодер получает только нормальные данные технологических процессов и минимизирует ошибку реконструкции, то есть разницу между исходными и восстановленными данными. После завершения обучения ему предоставляется новый набор данных, в котором могут присутствовать аномалии.

2. Описание LSTM

Долгая краткосрочная память (*Long short-term memory; LSTM*) – особая разновидность архитектуры рекуррентных нейронных сетей, способная к обучению долговременным зависимостям. Они были представлены Зеппом Хохрайтер и Юргеном Шмидхубером в 1997 году, а затем усовершенствованы и популярно изложены в работах многих других исследователей. Они прекрасно решают целый ряд разнообразных задач и в настоящее время широко используются [1].

LSTM-алгоритмы специально созданы для решения проблемы долговременной зависимости. Они изначально способны запоминать информацию в течение длительного времени, а не вынуждены этому обучаться с трудом.

Нейросети LSTM используют в целях классификации, обработки и построения прогнозов на основе временных рядов, где взаимосвязанные явления имеют возможность происходить с неопределенной временной периодичностью.

LSTM имеет структуру, подобную цепочке, но внутренний модуль имеет другую структуру. Внутри вместо одного слоя их четыре, взаимодействующих друг с другом.

Любая рекуррентная нейронная сеть представляет собой последовательность повторяющихся модулей. В стандартной RNN каждый такой модуль имеет довольно простую структуру, например, это может быть один слой с активационной функцией $\tanh$ (гиперболический тангенс). Архитектура LSTM представлена на рис. 1.

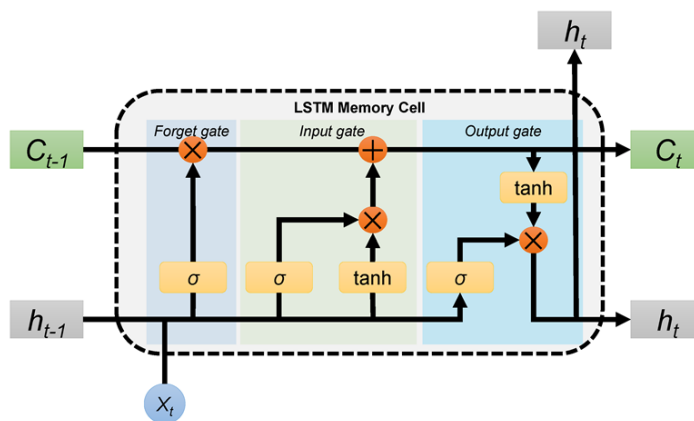


Рис. 1. Архитектура LSTM

Принцип работы рекуррентных нейронных сетей заключается в добавлении механизма памяти к искусственным нейронным сетям. Главная особенность LSTM-моделей связана с алгоритмом, позволяющим определять память как «короткую», что дает возможность на каждом этапе обучения обновлять хранимую информацию либо сразу, либо спустя несколько итераций. Это позволяет LSTM эффективно решать проблему долговременной зависимости, запоминая данные как на краткие, так и на длительные временные интервалы, что особенно важно при прогнозировании макроэкономических показателей. Такой эффект достигается благодаря отсутствию функции активации внутри рекуррентных компонентов, что предотвращает размывание значений во времени и устраняет проблему исчезающего градиента при обучении методом обратного распространения ошибки.

Основными элементами LSTM являются ячейки памяти и три вида фильтров: забывания, входной и выходной. Ячейка выполняет функцию хранения информации и передает её по всей цепочке модулей, что позволяет использовать даже ранние временные данные на поздних этапах, устраняя проблему кратковременной памяти.

Помимо прогнозирования, LSTM-модель также применяет графические и статистические методы для предварительного анализа временных рядов и интерпретации результатов. Полученные данные сравниваются с выводами других исследований и экспертными оценками для более точного анализа.

Результаты

1. Обработка данных

Для построения модели автоэнкодера необходим набор данных с оборудования. В качестве выборки будут три датчика температуры компрессора, для которых будет и проводиться анализ. Данные были загружены в Excel-файл за 4 месяца. После чего необходимо обработать выборку, т.к. у каждого собранного значения датчика своя временная метка. В общем итоге получаются данные по трем датчикам, у которых получились разные длины и разное время сбора. Необходимо обработать данные, тем самым привести все данные к общей временной метки, а также убрать пропуски.

Для ускорения сходимости модели важно предварительно масштабировать данные, так как большие значения во входных параметрах могут замедлять процесс обучения. Для нормализации использовался `StandardScaler` из библиотеки `sklearn` [2].

`Sklearn` настраивался исключительно на обучающей выборке, после чего применялся для преобразования тестовых данных, которые ранее не использовались в процессе обучения. Если бы нормализация выполнялась на всем наборе данных сразу, это могло бы улучшить результаты модели на имеющихся данных, но снизить её способность к обобщению, что негативно сказалось бы на работе с новыми реальными данными.

После масштабирования необходимо преобразовать данные в формат для моделирования LSTM с помощью библиотеки `MinMaxScaler`:

```
scaler = MinMaxScaler() #StandardScaler()
X_train_scaled = scaler.fit(X_train).transform(X_train)
X_train_scaled = pd.DataFrame(X_train_scaled, columns = X_train.columns, index = X_train.index)
X_test_scaled = scaler.transform(X_test)
X_test_scaled = pd.DataFrame(X_test_scaled, columns = X_test.columns, index = X_test.index)
```

2. Обучение модели автоэнкодера

Перед обучением модели необходимо инициализировать модель, выставить базовые настройки: количество эпох, слоёв, параметры сети и размер пакета.

Пример инициализации:

```
model=Sequential()
model.add(Dense(10,activation=act_func,
kernel_initializer='glorot_uniform',
kernel_regularizer=regularizers.l2(0.0),
input_shape=(X_train_scaled.shape[1])))
model.compile(loss='mae',optimizer='adam') # функция потерь, оптимизатор
# количество эпох, размер пакета
NUM_EPOCHS=20
BATCH_SIZE=10
```

После чего обучаем модель и производим распределение функции потерь в обучающем наборе. Строим график распределения вычисленных потерь в обучающем наборе (рис. 2).

Его можно использовать для определения подходящего порогового значения для выявления аномалии. При этом можно убедиться, что этот порог установлен выше «уровня шума», и что любые отмеченные аномалии должны быть статистически значимыми выше шумового фона.

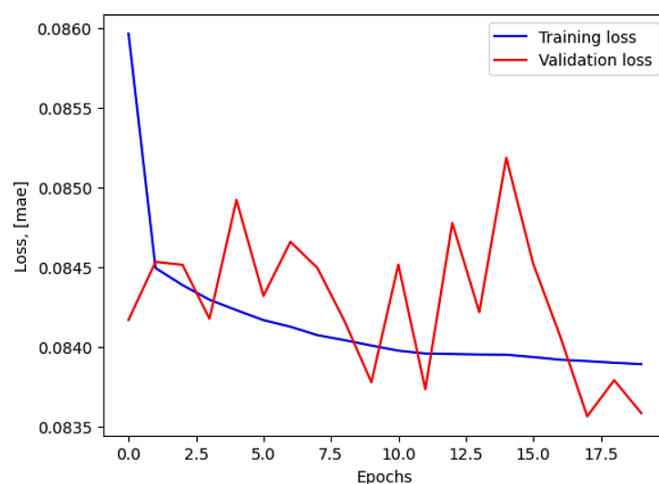


Рис. 2. График распределения вычисленных потерь

На графике представлены две линии: «Training loss» – ошибка на выборке для обучения и «Validation loss» – ошибка на выборке для тестирования.

Определим порог для определения аномалии, где выставляем самое большое число ошибки (квантиль): $\text{threshold} = \text{scored}['\text{Loss_mae}']. \text{quantile}(q=0.999)$.

Исходя из приведенного выше распределения потерь и порога рассчитаем потери на тестовом наборе. Вклад параметра в аномалию (доля ошибки и прирост доли) или по-другому – насколько сильно каждая точка влияет на то, была ли аномалия или нет (в процентах). Это и будет результат модели автоэнкодера (рис. 3).



Рис. 3. Результат работы автоэнкодера

Подводя итог, можно сказать, что с помощью модели автоэнкодера можно определить аномалии по собранным данным с оборудования, то есть, как сильно/не сильно значение с датчика влияет на возникновение аномалии. На рис. 3 произведена визуализация полученной средней абсолютной ошибки (Loss_mae) и сглаженной ошибки (Loss_mae_EWMA), а также граница (если ошибка выше, чем порог – это аномалия). На графике ниже представлены значения влияния значений датчиков на аномалию.

3. Линейный прогноз

Линейный прогноз – это метод предсказания будущих значений временного ряда на основе линейной зависимости между прошлыми и будущими данными. Он основан на модели линейной регрессии, где будущие значения представляются как линейная комбинация предыдущих наблюдений. Такой подход эффективен при наличии устойчивых трендов, но плохо учитывает сложные нелинейные зависимости.

Линейный прогноз в LSTM используется для сравнения качества предсказаний, выявления трендов и создания гибридных моделей, комбинирующих линейные и нелинейные зависимости.

Линейная регрессия описывается уравнением:

$$y = b_0 + b_1x_1 + b_2x_2 + b_nx_n + \varepsilon,$$

где:

- y – предсказанное значение,
- $x_1, x_2, \dots, x_n$ – независимые переменные,
- b_0 – свободный член,
- $b_1, b_2, \dots, b_n$ – коэффициенты (веса), определяющие влияние факторов,
- ε – ошибка (шум).

Линейная регрессия применяется для прогнозирования и анализа влияния факторов на результат, а также как базовый алгоритм машинного обучения.

Воспользуемся библиотекой `sklearn` – `LinearRegression`. Результат моделирования представлен на рис. 4. Сверху представлена граница, слева – данные для обучения (реальные), справа – представление линейной регрессии, то есть предсказание, как дальше пойдет линия с точками.

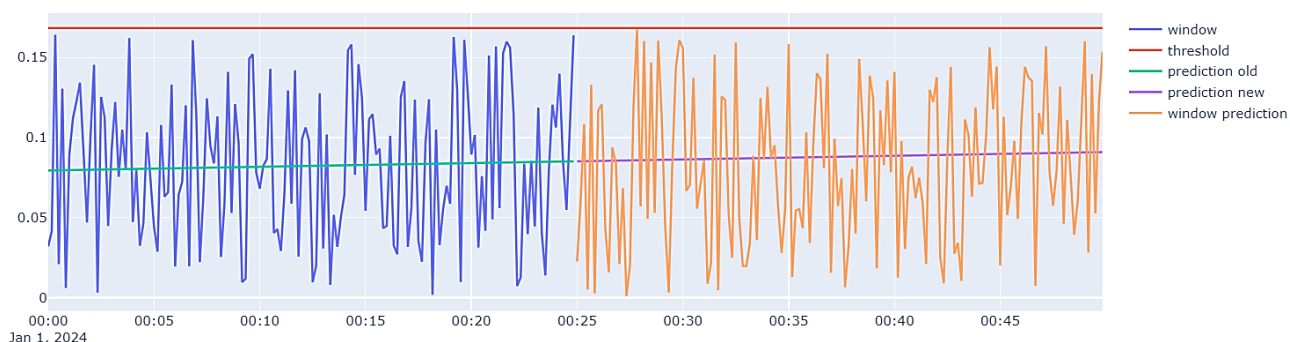


Рис. 4. Результат линейной регрессии

4. Модель LSTM

Для обучения модели данные были разбиты на обучающую и тестовую выборки, для обучения использовались 80%, для тестирования – 20%. После чего преобразуем длинную последовательность данных во множество более коротких последовательностей (по 60 тактов на последовательность), которые сдвинуты на один такт.

Далее была построена модель машинного обучения LSTM (рис. 5), состоящая из 4 слоев:

```
model = Sequential()
model.add(layers.LSTM(100, return_sequences=True, input_shape=(X.shape[1], 1)))
model.add(layers.LSTM(50, return_sequences=False))
model.add(layers.Dense(25))
model.add(layers.Dense(1))
model.summary()
```

Layer (type)	Output shape	Param #
lstm (LSTM)	(None, 60, 100)	40,800
lstm_1 (LSTM)	(None, 50)	30,200
dense_4 (Dense)	(None, 25)	1,275
dense_5 (Dense)	(None, 1)	26

Рис. 5. Модель LSTM

Затем данные для тестирования были также приведены к виду последовательностей. Полученные предсказания на данных для тестирования и фактические значения представлены на рис. 6. Ошибка предсказания RMSE составила 0.0015, что является отличным результатом. Исходя из произведённого опыта, можно подтвердить, что LSTM предсказывает поведения автоэнкодера и учится на полученных аномалиях, чтобы их предсказывать.

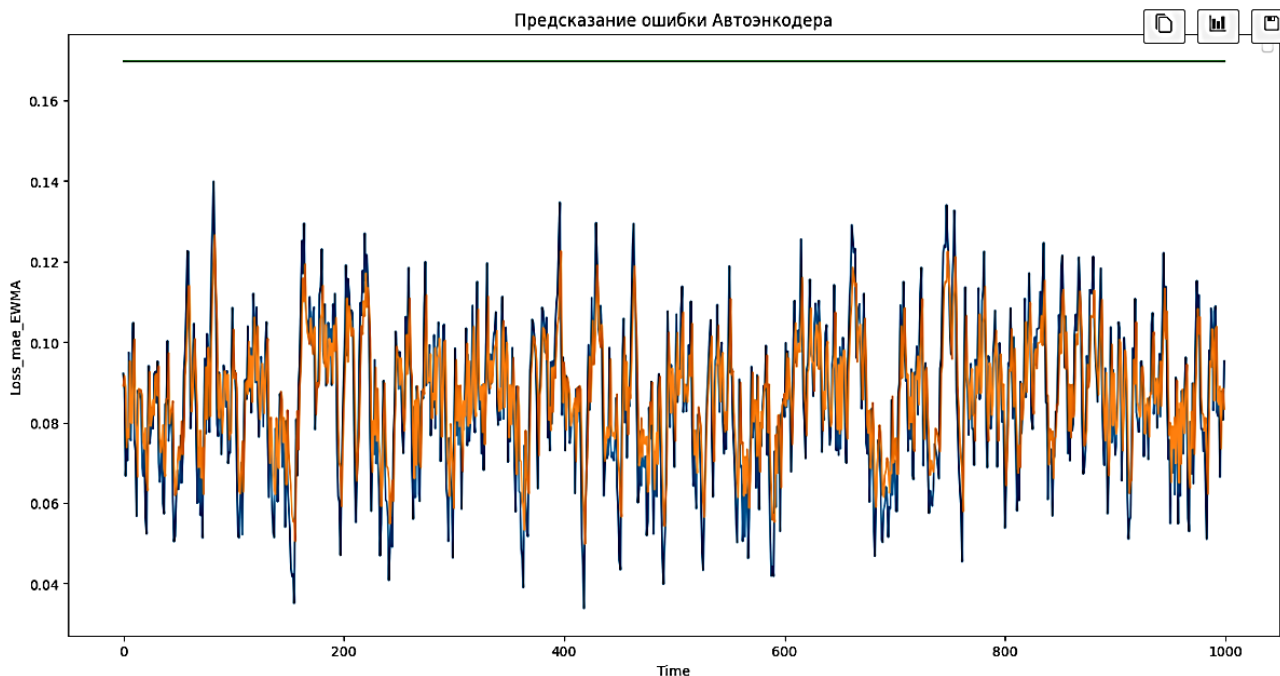


Рис. 6. Предсказание ошибки автоэнкодера

Выводы и заключение

Для построения модели автоэнкодера были использованы данные с датчиков температуры компрессора, с предварительной обработкой данных (масштабирование, синхронизация временных меток). Модель автоэнкодера успешно обучалась, минимизируя ошибку реконструкции, и позволяла определить порог для аномалий. Результаты демонстрируют, как данные датчиков могут влиять на обнаружение аномалий.

В модели LSTM использовалась линейная регрессия для сравнения предсказаний и выявления трендов, после чего был построен более сложный подход на основе LSTM. Она была обучена на 80% данных, а тестирование показало хорошую точность с ошибкой предсказания $RMSE = 0.0015$, что подтверждает эффективность использования LSTM для предсказания на основе временных рядов.

Список источников

1. Долгая краткосрочная память. URL: https://ru.wikipedia.org/wiki/Долгая_краткосрочная_память (дата обращения: 12.01.2025).
2. Scikit-learn. URL: <https://en.wikipedia.org/wiki/Scikit-learn> (date accessed: 11.02.2025).
3. Кузьмина М. Г. Многослойные сети-автоэнкодеры в задачах анализа и обработки гиперспектральных изображений. URL: https://keldysh.ru/papers/2021/prep2021_28.pdf (дата обращения: 12.02.2025).
4. Смирнов А. В., Тищенко И. П. Применение нейронных сетей сиамской архитектуры в задачах классификации продуктов различных категорий на прилавках универсама // Программные системы: теория и приложения. 2024. Т. 15, № 2 (61). С. 113–137.