

Формирование датасета и обучение моделей CNN+LSTM

1. Формирование датасета

Исходные данные:

- Временной ряд тока экструдера (`ter value_14`) и другие параметры (`value_10`, `value_16`, `value_48`, `mode`)
- Период: 05.03.2024 – 05.03.2025
- Объём: 685 201 значение

Этапы подготовки:

1. Добавление признаков:

- 10 **лаговых значений** тока (`value_14_lag_1 ... value_14_lag_10`)
- **Разность** между текущим и предыдущим значением (`value_14_diff_1`)
- Итого: **160 параметров** → затем отобрали ключевые

2. Сегментация по режиму работы:

- Данные разбиты на **сегменты**, где `mode = 1` (рабочий режим)
- Получено **1687 сегментов**
- Удалены сегменты короче 50 точек → осталось **1397 сегментов**

3. Стандартизация и выравнивание:

- Все сегменты приведены к длине **360 точек**:
 - Короткие — дополнены нулями справа
 - Длинные — обрезаны слева
- Данные **стандартизированы** (`mean=0`, `std=1`)

4. Разделение выборки:

- Обучающая / тестовая = **80% / 20%**

2. Архитектура моделей: CNN + LSTM

Общая структура:

[Conv1D → MaxPooling1D] → [LSTM] → [Dense]

Базовая модель:

- **Conv1D**: 128 фильтров, `kernel_size=3`, активация ReLU
- **MaxPooling1D**: `pool_size=2`
- **LSTM**: 128 нейронов, `return_sequences=False`
- **Dropout**: 0.2
- **Dense**: выходной слой + регуляризация (L2)

- **Loss:** mean_squared_error (MSE)
 - **Оптимизатор:** RMSprop (lr=0.001)
-

3. Эксперименты и настройка

Параметры обучения:

- **Размер окна (window_size):** 20, 30, 40, 50, 60
- **Горизонт прогноза (forecast_horizon):** 10, 20, 30, 40, 50, 60
- **Лучшая комбинация:** window_size=20, forecast_horizon=10
→ На других комбинациях **значительно хуже качество**

Тестировались изменения:

Изменение	Эффект
Увеличение фильтров в Conv1D до 256	Незначительное улучшение
Увеличение LSTM-нейронов до 256	Высокие потери на первых эпохах
Добавление второго LSTM-слоя	Увеличение времени обучения, нет прироста
Смена оптимизатора с Adam на RMSprop	Снижение потерь на первой эпохе
Добавление L2-регуляризации	Помогает против переобучения

4. Результаты и выводы

Лучшая модель:

```
model.add(Conv1D(filters=128, kernel_size=3, activation='relu', input_shape=input_shape))
model.add(MaxPooling1D(pool_size=2))
model.add(LSTM(units=128, return_sequences=False))
model.add(Dropout(0.2))
model.add(Dense(units=128, activation='relu', kernel_regularizer=l2(0.01)))
model.add(Dropout(0.2))
model.add(Dense(units=n_outputs))
```

Ключевые результаты:

Горизонт прогноза	MSE (на тесте)
10	1492.18
20	2902.12
30	4191.38

♦ Чем дальше горизонт — тем хуже качество (ожидаемо).

5. Основные выводы

1. Датасет требует пересмотра:

- Несмотря на разные архитектуры, **MSE остаётся высокой**
- Изменения в модели **не дают существенного улучшения**
- Вероятная причина — **недостаточное качество или структура данных**

2. Проблемы с данными:

- Сегменты разной природы (пуск, работа, простой) могут мешать обучению
- Нулевые значения и пропуски искажают сигнал
- Текущая сегментация (mode=1) может быть **недостаточно точной**

3. Рекомендации:

- Пересмотреть **логику формирования датасета**
- Уточнить у заказчика **критерии рабочего режима**
- Рассмотреть **разделение на подрежимы** (выход на режим, стабильная работа)
- Попробовать **модели с механизмом внимания** для фокусировки на ключевых участках