

# Выбор архитектуры модели для прогнозирования тока экструдера

## 1. Цель проекта

Разработать **точную и устойчивую модель прогнозирования тока экструдера** на основе временных рядов с промышленного оборудования.

Ключевая задача — предсказание поведения в **реальном режиме работы**, с учётом нестационарности, шумов и смены режимов.

## 2. Эволюция подхода: от LSTM к гибридной модели с вниманием

Этап	Архитектура	Цель
1	LSTM(128) → LSTM(64) → Dense	Базовая модель, проверка применимости RNN
2	CNN → LSTM → Dense	Добавление локальных паттернов через свёртки
3	CNN → BiLSTM → Attention → GRU → Dense	Гибридная модель с механизмом внимания
4	Автоподбор гиперпараметров (Keras Tuner)	Оптимизация архитектуры и параметров

## 3. Подготовка данных

### Источники:

- Теги: `value_14` (ток экструдера), `value_10`, `value_16`
- Период: 05.03.2024 – 05.03.2025
- Частота: 2–6 измерений в минуту

### Этапы обработки:

- Восстановление временного индекса** — заполнение пропущенных секунд
- Линейная интерполяция** — для восстановления пропусков
- Формирование режима работы:**
  - `mode = 1`, если `value_14 != 0` (оборудование в работе)
  - `mode = 0`, если `value_14 == 0` (простой)
- Инжиниринг признаков:**
  - Лаги (1–6)
  - Разность между значениями

- Скользящие средние, СКО, дисперсия
  - Автокорреляция
  - Итого: до 48 признаков
- 


## 4. Этап 1: Базовая LSTM-модель

### Архитектура:

```
model = Sequential([
    Input(shape=(sequence_length, num_features)),
    LSTM(128, return_sequences=True),
    Dropout(0.2),
    LSTM(64),
    Dropout(0.2),
    Dense(1)
])
```

- **Оптимизатор:** Adam
- **Loss:** MSE
- **sequence\_length:** 10, 15, 20, 25

### Результаты (прогноз на 1 лаг):

Период обучения	MAE (обучение)	MAE (тест)
2 месяца	6.57	8.86
4 месяца	7.60	7.51 

 **Вывод:** 4 месяца — лучший баланс между обобщением и переобучением.

---

## 5. Этап 2: Гибридные модели с вниманием

### Цель:

Улучшить качество за счёт:

- **CNN** — выделение локальных аномалий
- **BiLSTM** — захват двунаправленных зависимостей
- **Внимание (Attention)** — фокусировка на критических участках
- **GRU** — компактная альтернатива LSTM

### Архитектура (на примере Модели №2):

```
inputs = Input(shape=(look_back, n_features))
```

```
# CNN: локальные паттерны
```

```
conv = Conv1D(filters=128, kernel_size=5, padding='causal', activation='relu')(inputs)
```

```
pool = MaxPooling1D(pool_size=2)(conv)

# BiLSTM: долгосрочная память
lstm = Bidirectional(LSTM(128, return_sequences=True))(pool)
drop1 = Dropout(0.3)(lstm)

# Attention: фокус на важных шагах
attention = Attention()([drop1, drop1])
attention = GlobalAveragePooling1D()(attention)

# GRU: дополнительная рекуррентная обработка
gru = GRU(160)(drop1)

# Объединение и выход
combined = Concatenate()([attention, gru])
dense = Dense(128, activation='relu', kernel_regularizer=l2(0.01))(combined)
drop2 = Dropout(0.3)(dense)
output = Dense(n_output_steps)(drop2)

model = Model(inputs, output)
```

---



## 6. Этап 3: Подбор архитектуры и параметров

### Тестировались:

- Разные фильтры в Conv1D (64–512)
- Размер ядра: 3, 5, 7
- Количество юнитов в LSTM/GRU
- Dropout: 0.1–0.5
- Learning rate: 1e-4 – 1e-2
- sequence\_length: 10, 15, 20, 25

### Ключевые выводы:

1. **Лучшие результаты** — при sequence\_length = 20
  2. Увеличение сложности не всегда улучшает качество
  3. **Модели №2 и №4** показали наилучший потенциал
  4. **Внимание и GRU** — критически важны для фокусировки на аномалиях
-

## 7. Этап 4: Автоматический подбор гиперпараметров (Hyperparameter Tuning)

### Инструмент:

- Keras Tuner (Bayesian Optimization)

### Цель:


Найти оптимальные параметры для **Моделей №1, №2, №3, №4**

(обучение прервано из-за лимитов Colab, но получены промежуточные результаты)

### Лучшие гиперпараметры (на примере Модели №2):

Параметр	Оптимальное значение
conv_filters	416
conv_kernel_size	5
lstm_units	64
gru_units	304
dense_units	424
dropout_rate	0.1
l2_reg	0.062
learning_rate	0.00038

## 8. Результаты: Сравнение моделей

Модель	MAE(обучение)	MAE (тест)	MEDIAN RE (тест)
LSTM (базовая)	7.60	7.51	—
CNN+LSTM (Справка 4)	38.7	136.43	—
Модель №1_авто	32.80	41.60	13.52%
Модель №2_авто 	<b>31.14</b>	<b>41.05</b>	<b>13.96%</b>
Модель №3_авто	31.18	41.23	14.53%
Модель №4_авто	34.04	43.36	15.18%

 **Лучшая модель: №2\_авто** — гибрид CNN + BiLSTM + Attention + GRU

## 9. Прогноз на разные горизонты

Горизонт	MAE (тест)
10 мин (20 лагов)	<b>136.43</b>
30 мин (60 лагов)	139.96
60 мин (120 лагов)	147.28

- ◆ Качество снижается с ростом горизонта — ожидаемо.
- ◆ Прогноз на **10–30 минут** — наиболее точен.



## 10. Итоговые выводы

1. **Базовая LSTM** — хороша для **краткосрочного прогноза (1 лаг)**, но не хватает выразительной силы для сложных паттернов.
2. **Гибридные модели с вниманием** — **наиболее эффективны** для данного типа данных:
  - CNN выделяет локальные аномалии
  - BiLSTM захватывает долгосрочные зависимости
  - Внимания фокусируется на критических участках
  - GRU улучшает динамику
3. **Автоподбор гиперпараметров** подтвердил:
  - Оптимальные значения фильтров, юнитов, dropout
  - **Модель №2** — лучший компромисс между качеством и сложностью
4. **Качество данных** — **ключевое ограничение**:
  - Смешанные режимы (пуск, работа, простой)
  - Нулевые значения
  - Отрицательная скорость → Требуется **уточнения у заказчика** и **улучшения сегментации**
5. **Дальнейшие шаги**:
  - Дообучить **Модель №2** с найденными гиперпараметрами
  - Провести **интерпретацию внимания** (SHAP, attention weights)
  - Реализовать **онлайн-прогнозирование**
  - Перейти к **прогнозированию отказов** (классификация)