****

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА  
(САМАРСКИЙ УНИВЕРСИТЕТ)»

ИНСТИТУТ ИНФОРМАТИКИ И КИБЕРНЕТИКИ

Кафедра программных систем

Дисциплина

Нейронные сети

ОТЧЕТ

по лабораторной работе №6

"Предсказание отказов оборудования"

Студент: Соколова А.Д.

Группа: 6301-020302D

Проверил:

профессор Тюгашев А.А.

Дата: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Самара  
2025

СОДЕРЖАНИЕ

[ПОСТАНОВКА ЗАДАЧИ 3](#_Toc197905845)

[1 Исходный текст программы 4](#_Toc197905846)

[2 Протокол исполнения 6](#_Toc197905847)

[ЗАКЛЮЧЕНИЕ 7](#_Toc197905848)

# ПОСТАНОВКА ЗАДАЧИ

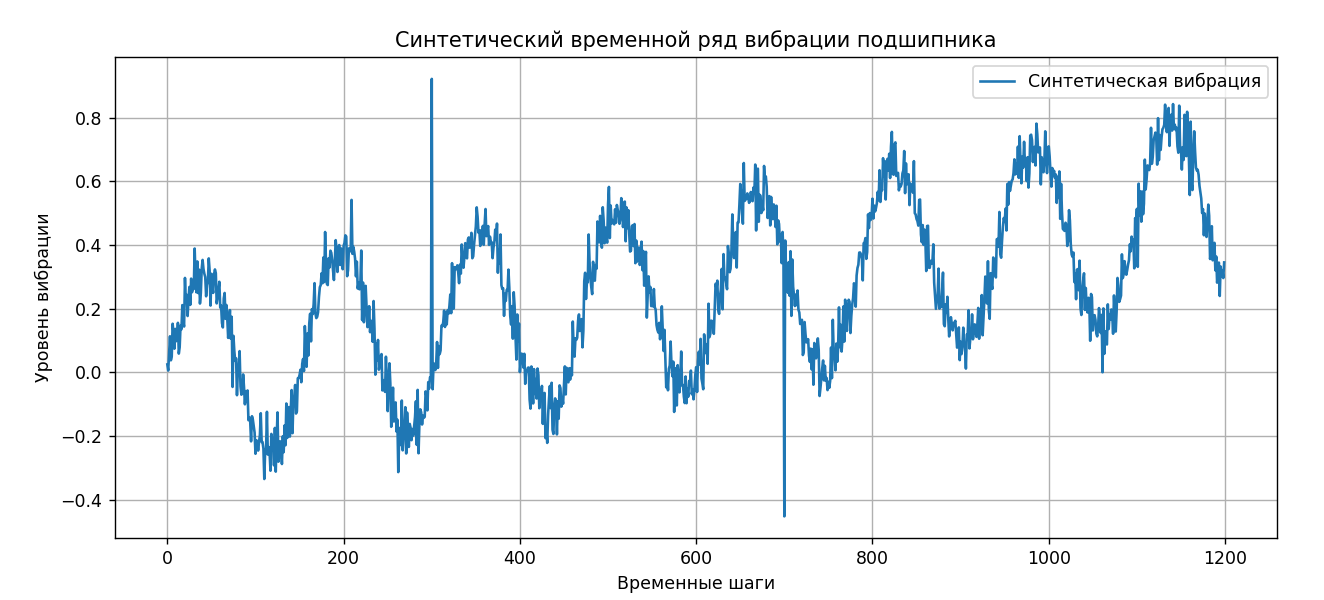
Цель работы построение модели нейронной сети, способную анализировать временные ряды, связанные с работой промышленного оборудования. В качестве примера используется временной ряд уровня вибрации подшипника.

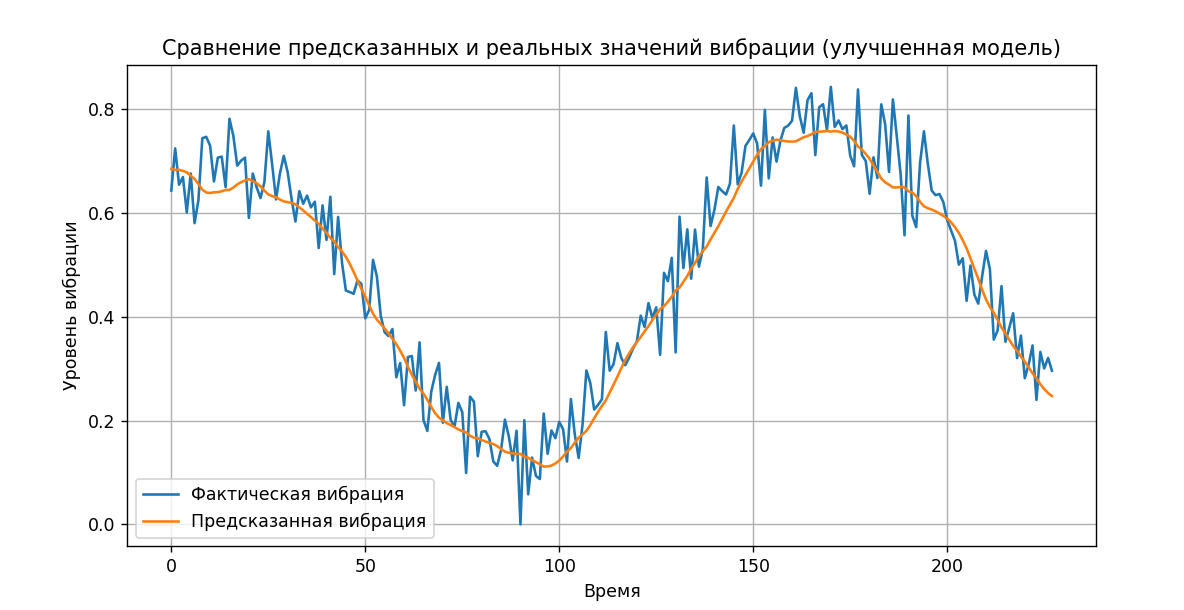
В рамках выполнения работы предполагается выполнение следующих задач:

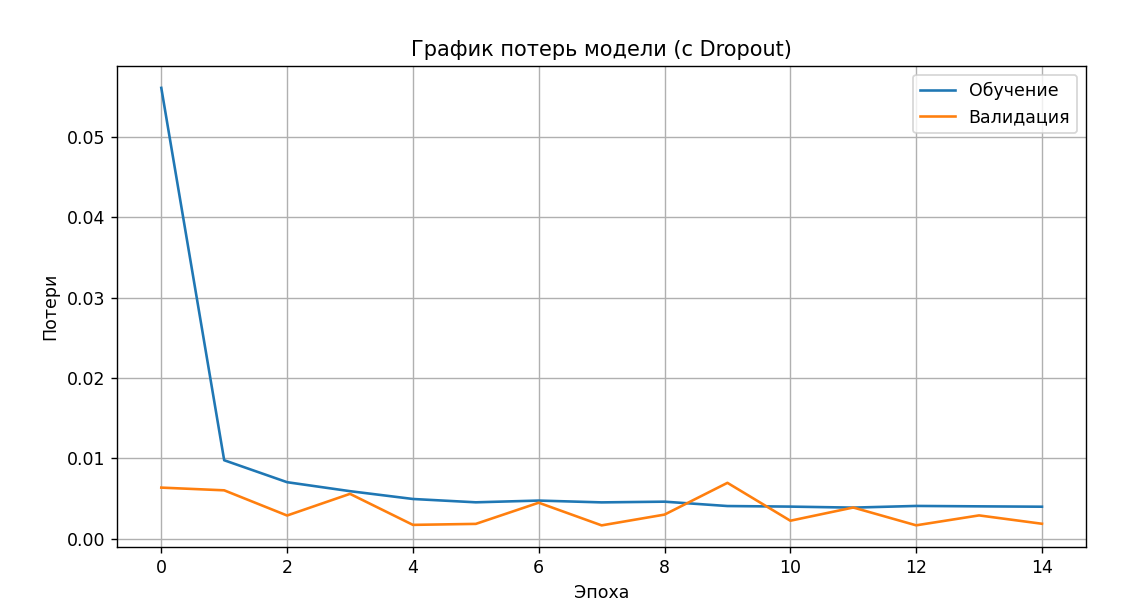
1. генерация, нормализация и преобразование данных;
2. обучение модели и оценка её точности на тестовых данных;
3. визуализация результатов и интерпретация поведения модели в процессе обучения.
4. Исходный текст программы

import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
from sklearn.preprocessing import MinMaxScaler  
from sklearn.metrics import mean\_squared\_error, mean\_absolute\_error, r2\_score  
from tensorflow.keras.models import Sequential  
from tensorflow.keras.layers import Dense, LSTM, Dropout  
  
np.random.seed(42)  
time\_steps = 1200  
base\_signal = 0.3 \* np.sin(0.04 \* np.arange(time\_steps))  
trend = np.linspace(0, 0.5, time\_steps)  
noise = 0.05 \* np.random.normal(0, 1, time\_steps)  
spikes = np.zeros(time\_steps)  
spikes[300] = 1.0  
spikes[700] = -0.8  
vibration = base\_signal + trend + noise + spikes  
df = pd.DataFrame({'Vibration': vibration})  
  
*# визуализация синтетического временного ряда вибрации*plt.figure(figsize=(12, 5))  
plt.plot(df['Vibration'], label='Синтетическая вибрация')  
plt.title('Синтетический временной ряд вибрации подшипника')  
plt.xlabel('Временные шаги')  
plt.ylabel('Уровень вибрации')  
plt.legend()  
plt.grid(True)  
plt.show()  
  
*# нормализация*scaler = MinMaxScaler()  
scaled\_data = scaler.fit\_transform(df)  
  
*# создание обучающей выборки*def create\_dataset(data, time\_step=60):  
 X, y = [], []  
 for i in range(len(data) - time\_step - 1):  
 X.append(data[i:i+time\_step, 0])  
 y.append(data[i+time\_step, 0])  
 return np.array(X), np.array(y)  
  
time\_step = 60  
X, y = create\_dataset(scaled\_data, time\_step)  
  
split = int(len(X) \* 0.8)  
X\_train, X\_test = X[:split], X[split:]  
y\_train, y\_test = y[:split], y[split:]  
X\_train = X\_train.reshape(-1, time\_step, 1)  
X\_test = X\_test.reshape(-1, time\_step, 1)  
  
model = Sequential()  
model.add(LSTM(64, return\_sequences=True, input\_shape=(time\_step, 1)))  
model.add(Dropout(0.2))  
model.add(LSTM(64, return\_sequences=False))  
model.add(Dropout(0.2))  
model.add(Dense(32, activation='relu'))  
model.add(Dense(1))  
model.compile(optimizer='adam', loss='mean\_squared\_error')  
  
*# обучение*history = model.fit(X\_train, y\_train, epochs=15, batch\_size=32, validation\_data=(X\_test, y\_test), verbose=1)  
  
*# график потерь*plt.figure(figsize=(10, 5))  
plt.plot(history.history['loss'], label='Обучение')  
plt.plot(history.history['val\_loss'], label='Валидация')  
plt.title('График потерь модели (с Dropout)')  
plt.xlabel('Эпоха')  
plt.ylabel('Потери')  
plt.legend()  
plt.grid(True)  
plt.show()  
  
*# предсказание*predicted = model.predict(X\_test)  
predicted = scaler.inverse\_transform(predicted)  
y\_test\_real = scaler.inverse\_transform(y\_test.reshape(-1, 1))  
  
*# график предсказаний*plt.figure(figsize=(10, 5))  
plt.plot(y\_test\_real, label='Фактическая вибрация')  
plt.plot(predicted, label='Предсказанная вибрация')  
plt.title('Сравнение предсказанных и реальных значений вибрации (улучшенная модель)')  
plt.xlabel('Время')  
plt.ylabel('Уровень вибрации')  
plt.legend()  
plt.grid(True)  
plt.show()  
  
*# метрики*mse = mean\_squared\_error(y\_test\_real, predicted)  
mae = mean\_absolute\_error(y\_test\_real, predicted)  
r2 = r2\_score(y\_test\_real, predicted)  
print(f"MSE: {mse:.4f}")  
print(f"MAE: {mae:.4f}")  
print(f"R²: {r2:.4f}")

1. Протокол исполнения

  
Рисунок 1 – Синтетический временной ряд

  
Рисунок 2 – Сравнение реальных и предсказанных значений

  
Рисунок 3 – График потерь модели

# ЗАКЛЮЧЕНИЕ

В рамках лабораторной работы была реализована система анализа временного ряда, моделирующего уровень вибрации подшипника промышленного оборудования. Использование LSTM позволило создать модель, способную предсказывать значения на основе прошлых измерений. Построенные графики продемонстрировали качество предсказаний и динамику потерь. Вычисленные метрики (MSE и MAE) подтвердили базовую точность модели. Такой подход может быть использован в задачах предиктивной диагностики для выявления потенциальных неисправностей.