CNC Milling Performance Analysis and Fault Detection

Tool Wear Prediction Using LSTM

1. Introduction

This project focuses on predicting tool wear conditions in a CNC machining process using time-series data and deep learning models. The dataset includes various machining parameters, and the model aims to predict three target variables:

- **Tool Condition** (Multi-class classification)
- **Machining Finalized** (Binary classification)
- **Passed Visual Inspection** (Binary classification)

We employ an **LSTM-based neural network** to capture temporal dependencies in the dataset and predict these outputs.

2. Dataset Description

2.1 Data Source

The dataset is loaded from `Merged_Final_Data.csv`.

2.2 Features Selection

The dataset originally had **56 columns**. After feature selection, **15 relevant features** were chosen:

- **`Z1_ActualPosition`**
- **`Z1_CommandPosition`**
- **`M1_sequence_number`**
- **`X1_OutputCurrent`**
- **`S1_ActualPosition`**
- **`S1_CommandPosition`**
- **`Y1_OutputCurrent`**
- **`S1_OutputCurrent`**
- **`clamp pressure`**
- **`feedrate`**
- **`M1_CURRENT_FEEDRATE`**
- **`S1_CurrentFeedback`**
- **`X1_DCBusVoltage`**
- **`X1_OutputVoltage`**
- **`Y1_DCBusVoltage`**

2.3 Target Variables

1. `tool condition` (Categorical, Multi-class)
2. `machining finalized` (Binary)
3. `passed_visual_inspection` (Binary)

2.4 Pre-processing Steps

- **Dropping unnecessary columns:** `Unnamed: 0`, `No`, `Machining Process`
- **Sorting data by timestamp** to ensure proper time-series structure.
- **Feature normalization** using `StandardScaler ()`.
- **Creating sequences** for LSTM model (sequence length = 10).
- **Splitting dataset** into `train (80%)` and `test (20%)`.

3. Model Architecture

We build a multi-output LSTM model:

- **Input Layer:** LSTM processes sequences of **(10 time steps, 15 features)**.

- **LSTM Layers:**
  - LSTM (128, return sequences=True)
  - LSTM (64)
- **Dropout Layer:** (0.2) to reduce overfitting.
- **Output Layers:**
  - `tool_condition_output`: Dense (3, activation='softmax')
  - `machining_finalized_output`: Dense (1, activation='sigmoid')
  - `passed_visual_inspection_output`: Dense(1, activation='sigmoid')

3.1 Model Compilation

- **Optimizer:** Adam
- **Loss functions:**
  - `sparse_categorical_crossentropy` for `tool condition`
  - `binary_crossentropy` for `machining finalized` & `passed_visual_inspection`
- **Metrics:** Accuracy

4. Training and Evaluation

4.1 Model Training

- **Epochs:** 50
- **Batch Size:** 32

```
model. Fit(X_train, {'tool_condition_output':
y1_train,'machining_finalized_output': y2_train,
'passed_visual_inspection_output': y3_train},
          validation_data=(X_test,
{'tool_condition_output': y1_test,
'machining_finalized_output': y2_test,
'passed_visual_inspection_output': y3_test}),
          epochs=50, batch_size=32)
```

4.2 Model Evaluation

```
eval_results = model.evaluate(X_test, {
    'tool_condition_output': y1_test,
    'machining_finalized_output': y2_test,
    'passed_visual_inspection_output': y3_test
})
print("Evaluation Results:", eval_results)
```

5. Conclusion

This project successfully applies LSTM for tool wear prediction, handling multiple outputs simultaneously. Further improvements can be made using hyperparameter tuning and advanced architectures.

Tool Wear Prediction Dashboard Documentation

Overview

This project involves predicting tool wear conditions using a trained LSTM-based deep learning model. The application provides a dashboard for real-time predictions based on user input. It predicts the following:

- **Tool Condition** (Good, Worn, Damaged)
- **Machining Finalization** (Yes/No)
- **Visual Inspection Pass/Fail**

Dataset Details

The dataset consists of multiple time-series features related to tool wear and machining processes. The key target variables are:

- `tool_condition`
- `machining_finalized`
- `passed_visual_inspection`

Model Details

- **Architecture:** LSTM-based multi-output model
- **Input Features:** 15 selected features from the dataset
- **Sequence Length:** 10 time steps
- **Outputs:** Multi-class classification for tool condition and binary classification for machining finalization & visual inspection

Preprocessing Steps

1. **Data Cleaning**: Unnecessary columns were removed.
2. **Feature Selection**: The most relevant 15 features were chosen.
3. **Scaling**: StandardScaler was applied to normalize data.
4. **Sequence Preparation**: Time-series sequences of length 10 were created.

Dashboard Implementation

- **Framework Used:** Streamlit
- **Input Handling:** User enters values for the selected 15 features.
- **Preprocessing:** Inputs are scaled using the same scaler used in training.
- **Prediction Process:**
    - The model takes the reshaped sequence input.
    - It outputs predictions for all three target variables.

    - The results are interpreted into human-readable format (Good/Worn/Damaged, Yes/No, Passed/Failed).
- **Visualization:** The dashboard dynamically displays the predicted values.

How to Run the Dashboard

1. Install dependencies:
2. `pip install streamlit tensorflow numpy pandas scikit-learn`
3. Save the trained model (`tool_wear_model.h5`) and scaler (`scaler.pkl`) in the working directory.
4. Run the Streamlit app:
5. `streamlit run app.py`
6. Enter feature values in the sidebar and click **Predict**.
7. The predictions will be displayed on the dashboard.

Expected Output

- **Tool Condition:** Good/Worn/Damaged
- **Machining Finalized:** Yes/No
- **Passed Visual Inspection:** Passed/Failed

Conclusion

This dashboard provides an interactive way to predict tool wear conditions in a machining process using deep learning. It can be extended with real-time data integration for automated monitoring.