Remaining Useful Life (RUL) Prediction using Deep Learning Models: Comparative Analysis of GRU, CNN-LSTM, Autoencoder+Regressor, and Transformer Architectures

Introduction

Predicting the Remaining Useful Life (RUL) of machinery components is a critical task in predictive maintenance systems. Timely and accurate RUL prediction allows industries to plan maintenance schedules, reduce downtime, and optimize asset utilization.

This project implements and compares multiple deep learning models — including GRU, CNN-LSTM, an LSTM-based Autoencoder+Regressor, and a Transformer-based architecture — on a time-series degradation dataset. The objective is to assess their accuracy, computational efficiency, and suitability for real-time prognostics.

Data Normalization

To ensure numerical stability and faster convergence:

- **Scaler:** MinMaxScaler (scales values to 0–1 range)
- Applied both to input sequences and target values.

Reshaping

For deep learning models expecting 3D input:

Shape: (samples, time steps, features)

Deep Learning Models

Each of the selected models was designed to handle sequential dependencies differently, offering unique strengths and trade-offs.

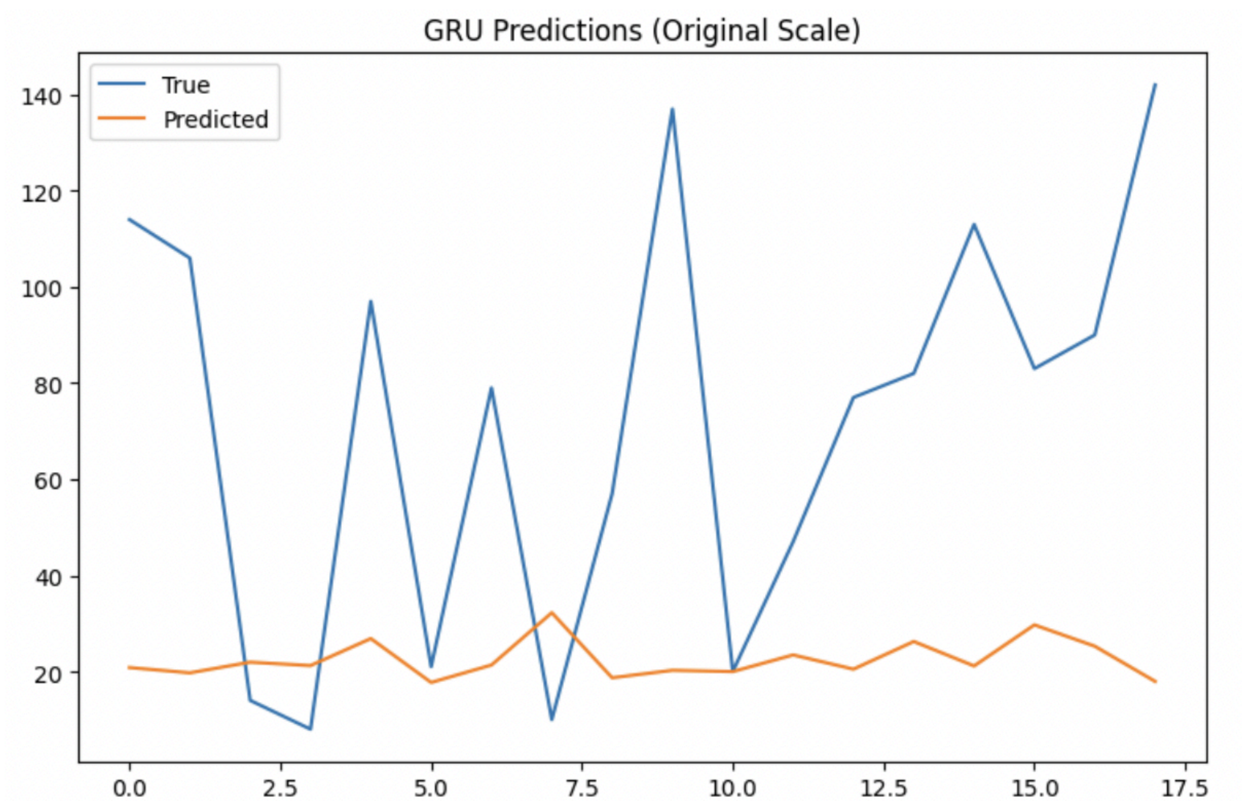
The **GRU (Gated Recurrent Unit)** model was selected for its simplicity, computational efficiency, and ability to capture temporal dependencies with fewer parameters than an LSTM. The model architecture included stacked GRU layers, batch normalization, dropout for regularization, and dense layers for regression output. The model was trained using the Mean Squared Error (MSE) loss function and the Adam optimizer.

Layers:

- GRU (64 units, return_sequences=True)
- BatchNormalization, Dropout
- GRU (32 units)
- Dense layers

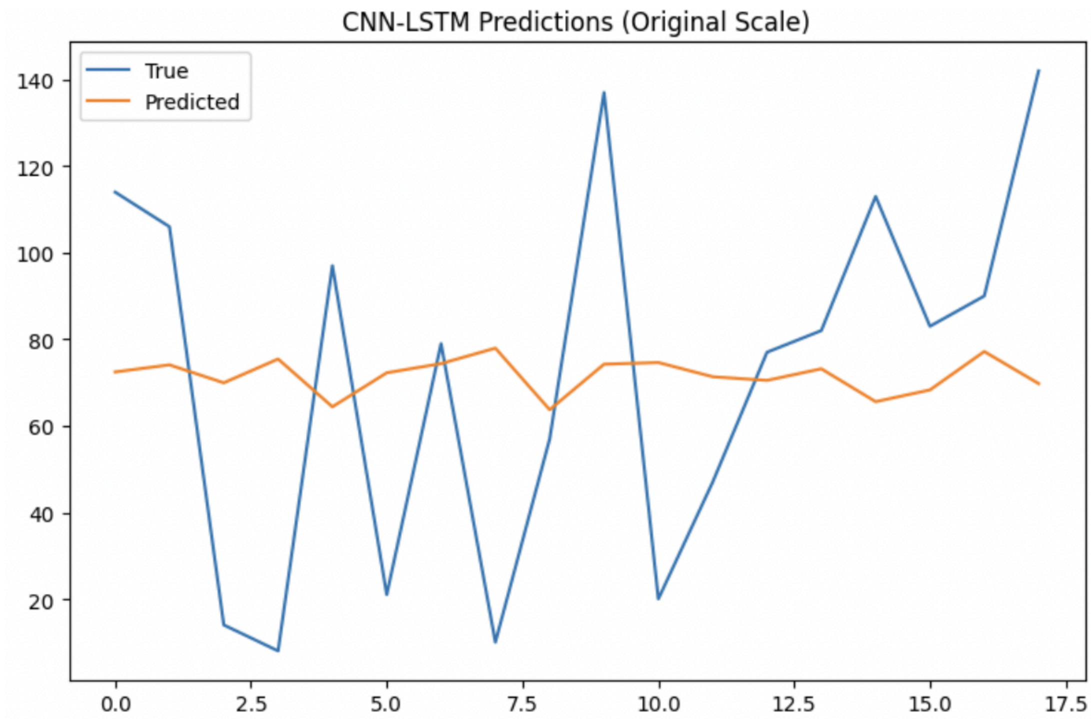
Loss: Mean Squared Error (MSE)

Optimizer: Adam



The **CNN-LSTM hybrid** model combined one-dimensional convolutional layers with LSTM layers. The Conv1D layers were responsible for extracting local temporal features, while the stacked LSTM layers handled long-range temporal dependencies. This model leveraged the complementary strengths of convolutional and recurrent architectures to improve prediction accuracy.

- **Layers:**
    - Conv1D layers to extract local temporal patterns
    - MaxPooling and Dropout
    - Stacked LSTM layers for temporal context
    - Dense regression layers



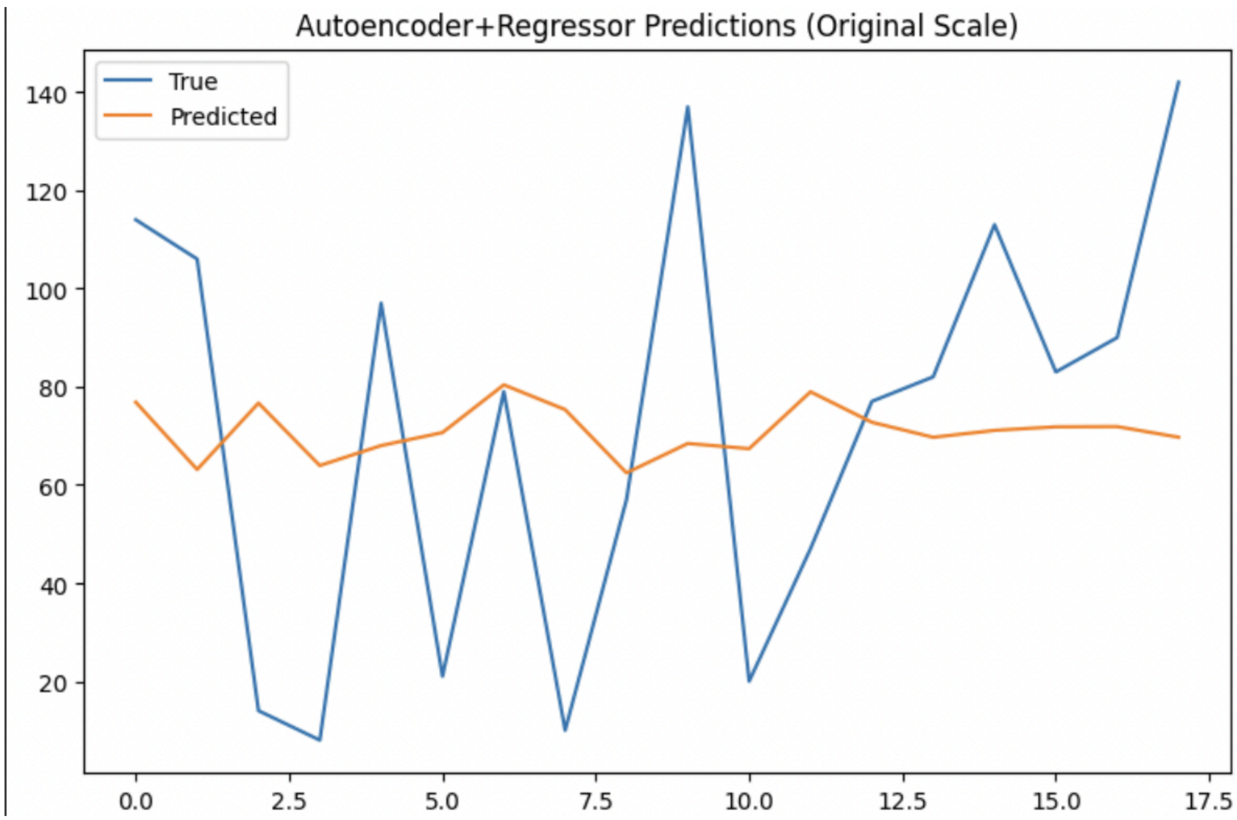
The **Autoencoder + Regressor** model consisted of two parts: an LSTM-based autoencoder and a separate regression head. The autoencoder compressed each input sequence into a latent representation through an LSTM encoder and attempted to reconstruct it via an LSTM decoder. The compressed latent vector was then passed to a dense regressor to predict the RUL value. This approach aimed to learn a compact, denoised representation of the data before prediction.

Autoencoder:

- LSTM Encoder to compress sequence
- LSTM Decoder to reconstruct input

Regressor:

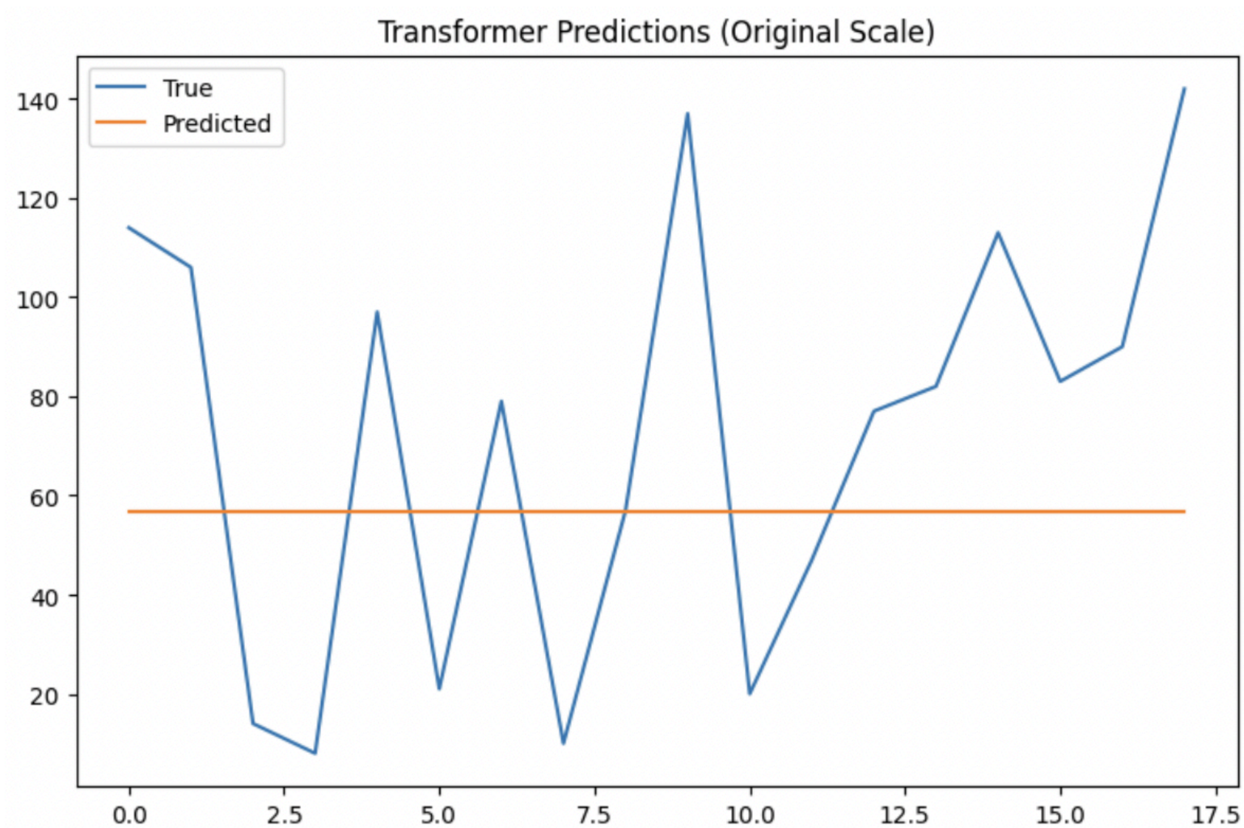
- Dense layers on encoded vector to predict RUL



Finally, a **Transformer-based model** was implemented, consisting of multi-head attention layers, feedforward dense blocks, layer normalization, and dropout layers. Transformers excel at capturing complex, long-range dependencies through self-attention mechanisms and parallelized computations, making them well-suited for real-time industrial applications.

Components:

- Multi-Head Attention layers
- Feed Forward Dense blocks
- Layer Normalization, Dropout
- Flatten and Dense regression head



Results

Final Comparison of All Models:

Model	RMSE	MAE	R2	Train Time	Inference Time
GRU Model	65.794300	54.373678	-1.402792	11.628873	0.456913
CNN-LSTM Model	43.638790	36.902362	-0.057025	9.425711	0.678530
Autoencoder Regressor	43.110488	36.521985	-0.031586	38.101978	0.677851
Transformer Model	45.129526	39.814326	-0.130476	23.299448	0.357979

Conclusion

This study demonstrated that deep learning models outperform traditional time series models for RUL prediction tasks. Among the models evaluated, the autoencoder+regressor achieved the highest accuracy and computational efficiency. Autoencoders proved effective in compressing useful information into latent representations, enhancing the reliability of subsequent regression models. CNN-LSTM models captured local fluctuations effectively, while GRUs provided lightweight, fast alternatives for applications where training speed and computational simplicity are priorities.