Final Project Paper

LSTM RNN Time Series Modeling of Fatigue

Authors: Geoffrey Hadfield  
Instructor: Dr. Shusen Pu  
Date: 2/9/2025

Abstract

In this project, we explore how deep learning—specifically long short-term memory (LSTM) recurrent neural networks—can be used to model fatigue from time series biomechanical data. Our work follows the Deep Learning Project Guidelines and builds on the techniques and concepts introduced during class. We developed an end-to-end pipeline that includes data loading, feature engineering, feature importance analysis, and model training using various RNN architectures. By incorporating class topics such as sequential processing, handling gradient challenges, and advanced architectures like GRUs and bidirectional networks, we demonstrate practical solutions to real-world problems in fatigue prediction. This paper outlines our methodology, discusses the experimental results, and reflects on the challenges and lessons learned.

1. Introduction

Fatigue modeling using time series data is an important challenge in fields like sports science, healthcare, and ergonomics. Our project’s goal is to predict fatigue levels using an LSTM-based recurrent neural network. In accordance with the project guidelines provided by Dr. Shusen Pu, we have designed our work to reflect a practical application of deep learning techniques learned in class. Each stage of our project—from preparing the dataset to deploying advanced RNN models—represents our independent contributions as well as our collective classroom learning.

Specifically, our project:

* Analyzes a time series dataset collected from biomechanical sensors.
* Uses deep learning techniques (LSTM, GRU, and bidirectional RNNs) to capture temporal dependencies.
* Incorporates training strategies such as gradient clipping, dropout regularization, and adaptive optimizers to improve model convergence.
* Employs feature importance analysis methods like permutation importance, recursive feature elimination (RFE), and SHAP to understand the influence of different features.
* Extends our analysis by training joint-specific injury risk models and showcasing a forecasting example.

In the following sections, we describe our background knowledge from class, our data preprocessing and feature engineering steps, our model design and training approach, and the experimental outcomes of our analysis.

2. Background and Related Work

2.1 Deep Learning Project Guidelines

Our project was designed in strict adherence to the guidelines provided by Dr. Shusen Pu. The guidelines required us to either select a dataset of personal interest or reproduce findings from a peer-reviewed study. We chose to work with a real-world time series dataset related to biomechanical fatigue. Key requirements included:

* Submitting a proposal detailing the dataset, deep learning methods to be used, and our initial Python proficiency.
* Demonstrating independent coding contributions.
* Providing a comprehensive report along with well-documented Python code.

2.2 RNN Fundamentals and Advanced Architectures from Class

In class, we learned that recurrent neural networks (RNNs) are ideal for processing sequential data because they maintain an internal state that helps the network remember past inputs. However, we also learned that basic RNNs struggle with long-term dependencies due to issues like vanishing and exploding gradients.

To address these challenges, our course covered:

* Long Short-Term Memory (LSTM) Networks:  
  LSTMs overcome the shortcomings of basic RNNs by using “gates” that decide what information to keep, update, or discard. This makes them especially useful for learning patterns over long sequences.
* Gated Recurrent Units (GRU):  
  GRUs simplify the LSTM design by combining some of the gating mechanisms, offering a more efficient model while still handling long-term dependencies effectively.
* Bidirectional RNNs:  
  These networks process data in both forward and backward directions, providing a more complete context when the entire sequence is available. This approach, however, is more suitable for offline analysis rather than real-time forecasting.

These topics were central to our project, and we applied the class concepts in every step of our pipeline.

3. Methodology

Our project pipeline is organized into several key stages, each reflecting a core concept from class.

3.1 Data Loading and Preparation

We began by loading the primary CSV file, which contains extensive sensor and biomechanical data, along with a JSON file containing participant information. In our data loading module, we added a unique participant ID and merged the participant details with the sensor data. This step ensured that our dataset was complete and properly labeled for further analysis.

3.2 Feature Engineering

Our feature engineering process was designed to capture both the physical and temporal aspects of fatigue:

* Joint Features:  
  We calculated joint angles and range-of-motion (ROM) metrics in a simplified manner, inspired by in-class discussions on using vector-based methods. We also aggregated joint energy and power across multiple sensors to create comprehensive features such as overall joint energy.
* Derived Metrics:  
  Building on classroom topics, we created additional features such as:
  + *Energy Acceleration:* Estimating how quickly joint energy changes over time.
  + *Ankle Power Ratio:* Comparing the ongoing power between the left and right ankles.
  + *Asymmetry Metrics:* Determining differences between left and right joint measurements.
  + *Exhaustion Rate and Simulated Heart Rate:* Designed to provide a dynamic measure of fatigue based on both exhaustion scores and energy levels.
* Temporal Features:  
  We applied windowing and lag techniques—concepts discussed during the course—to generate input sequences for our RNN models. Rolling statistics (such as moving averages and standard deviations) further helped capture short-term fluctuations in the data.

3.3 Feature Importance Analysis

To understand which features most influence fatigue and injury risk predictions, we applied several analysis techniques:

* Permutation Importance:  
  By randomly shuffling individual feature values and observing the drop in model performance, we identified features that were most critical to accurate predictions.
* Recursive Feature Elimination (RFE):  
  We iteratively removed the least important features until we identified a core subset that best supported the model.
* SHAP Analysis:  
  This method allowed us to explain our model’s predictions by showing the average contribution of each feature.

The consensus from these methods guided our selection of features for the final model training.

3.4 Model Architecture and Training

Based on class discussions and our feature analysis, we developed multiple models:

* Fatigue Prediction Model (Regression):  
  We built an LSTM model with one or more LSTM layers followed by dropout layers to help prevent overfitting. The final output layer produces a continuous value representing the fatigue score. We used adaptive optimizers (such as Adam) and incorporated gradient clipping to address potential training challenges.
* Injury Risk Prediction Model (Classification):  
  For predicting injury risk, our model architecture was similar to the fatigue model but used a sigmoid activation in the final layer to output a probability. This model was trained with binary cross-entropy loss.
* Joint-Specific Models:  
  We also trained individual models for specific joints to understand localized injury risks. These models were built using a subset of top features selected during the feature importance analysis.

3.5 Forecasting

Lastly, we implemented a forecasting function to predict future fatigue scores. By using the trained fatigue model to generate predictions from recent data sequences, we produced plots that visually compare the actual versus predicted time series. This forecasting approach not only demonstrates model performance but also reflects our understanding of practical sequence modeling techniques.

4. Experimental Results

4.1 Model Training Outcomes

Fatigue Model (Regression)

Our LSTM model for fatigue prediction showed rapid convergence during training. Early epochs demonstrated a steep drop in error, and after several epochs, both the training and validation errors reached very low levels. When additional metrics such as mean absolute error (MAE) were used, the performance was consistent with our expectations based on class exercises.

Injury Risk Model (Classification)

The injury risk classification model achieved high accuracy on both training and validation sets. Throughout the epochs, the validation accuracy typically exceeded 97%, confirming that our model was effective at predicting injury risk based on the engineered features.

Joint-Specific Models

While some joint-specific models (such as for the left ankle) initially started with lower accuracy, they improved over time. The variation in performance among different joints provides insight into the complexity and noise present in the biomechanical data, as discussed in class.

4.2 Feature Importance Insights

Our feature importance analysis consistently identified features like lagged exhaustion scores, rolling averages, and certain power ratios as the most influential. These findings are in line with class discussions on the importance of temporal features in time series modeling. The consensus ranking helped us narrow down a focused set of features for each model, reinforcing the practical application of the theoretical techniques we learned.

4.3 Forecasting Results

The forecasting function produced clear visualizations that compared actual fatigue trends with model predictions. The forecasts extended beyond the observed data, illustrating the model’s potential in making short-term predictions. These plots not only validate our model’s performance but also demonstrate our ability to deploy a complete end-to-end pipeline for time series forecasting.

5. Discussion

5.1 Application of Class Concepts

Our project is deeply rooted in the concepts taught during the course:

* Sequential Processing:  
  We applied the idea of processing data in sequences by creating time-windowed inputs for our LSTM models.
* Handling Gradients:  
  To address challenges like vanishing and exploding gradients, we implemented training strategies such as gradient clipping and dropout—techniques that were heavily emphasized in class.
* Advanced Architectures:  
  Although our primary focus was on LSTMs, our discussions and experiments also considered GRUs and bidirectional RNNs. These advanced architectures provided us with options to better capture context and long-term dependencies.
* Data Preprocessing and Feature Engineering:  
  The methods we used for scaling, windowing, and creating lag features are direct applications of the practical examples we covered during lectures.

5.2 Challenges and Future Work

While our results are promising, several challenges remain:

* Dataset Variability:  
  The inherent noise and variability in biomechanical data sometimes limited model performance. Future work could explore data augmentation techniques or more advanced imputation methods.
* Model Extensions:  
  Adding attention mechanisms or combining convolutional layers with RNNs might further improve our model’s ability to focus on the most relevant parts of the input sequence.
* Real-Time Forecasting:  
  Although our forecasting approach works well for offline analysis, transitioning to real-time predictions would require further optimization and perhaps model quantization techniques.

6. Conclusions

In this project, we demonstrated how LSTM-based RNNs can be effectively used to model fatigue using time series data. By carefully following the project guidelines and integrating the concepts learned in class, we developed a robust pipeline that spans data loading, feature engineering, feature importance analysis, and model training. Our results show that deep learning models—when designed with proper attention to sequential processing and gradient management—can accurately predict fatigue levels and injury risks. The project not only reinforces our understanding of RNNs and their advanced variants (such as GRUs and bidirectional networks) but also lays the groundwork for future improvements and real-world applications.

References

* Dr. Shusen Pu. Deep Learning Project Guidelines.
* Class lectures and lab sessions on Recurrent Neural Networks, LSTMs, GRUs, and time series forecasting.
* Relevant course materials and Python code examples discussed in class.