**SPORTS INJURY RISK PREDICTION BASED ON ENSEMBLE MODEL AND TIME SERIES MODEL**

|  |  |
| --- | --- |
| **School:** |  |
| **Faculty:** | Electronic and Information Engineering College |
| **Major:** | Electronic Information Engineering |
| **Class:** |  |
| **Group:** |  |
| **Member:** |  |

**2025-6**

**Contents**

[1. Introduction - 5 -](#_Toc201927103)

[2. Dataset - 5 -](#_Toc201927104)

[2.1 Data Source - 5 -](#_Toc201927105)

[2.2 Event Distribution - 6 -](#_Toc201927106)

[2.3 Collection Methods - 6 -](#_Toc201927107)

[3. Feature Description - 6 -](#_Toc201927108)

[3.1 Day Type - 6 -](#_Toc201927109)

[3.2 Week Type - 7 -](#_Toc201927110)

[3.3 Common Dataset Structure - 8 -](#_Toc201927111)

[3.4 Key Considerations - 8 -](#_Toc201927112)

[4. Data processing - 8 -](#_Toc201927113)

[4.1 Feature Name Standardization - 8 -](#_Toc201927114)

[4.2 Temporal Data Conversion and Sorting - 8 -](#_Toc201927115)

[4.3 Feature Filtering and Imputation - 8 -](#_Toc201927116)

[4.4 Challenge of Class Imbalance - 9 -](#_Toc201927117)

[4.5 SMOTE: Synthetic Minority Over-sampling Technique - 9 -](#_Toc201927118)

[4.5.1 Principle and Implementation - 9 -](#_Toc201927119)

[4.5.2 Advantages for Injury Prediction - 9 -](#_Toc201927120)

[4.6 SMOTE-ENN: Hybrid Over-sampling and Under-sampling - 9 -](#_Toc201927121)

[4.6.1 Overview - 9 -](#_Toc201927122)

[4.6.2 Synergistic Benefits - 10 -](#_Toc201927123)

[4.7 Data Processing Pipeline - 10 -](#_Toc201927124)

[4.7.1 Feature Integration - 10 -](#_Toc201927125)

[4.7.2 Resampling Workflow - 10 -](#_Toc201927126)

[4.7.3 Methodological Advantages - 10 -](#_Toc201927127)

[5. Feature Engineering for Temporal Pattern Recognition - 11 -](#_Toc201927128)

[5.1 Sliding Window Time-Series Metrics (LightGBM model) - 11 -](#_Toc201927129)

[5.2 Feature Space Transformation - 11 -](#_Toc201927130)

[5.3 Feature Importance Hierarchy - 11 -](#_Toc201927131)

[5.4 Temporal direction constraints - 11 -](#_Toc201927132)

[5.5 Modeling of the cumulative effect of damage risk and explanation of rationality - 12 -](#_Toc201927133)

[5.6 Limitations of Original Sliding Window Features (LightGBM model) - 12 -](#_Toc201927134)

[5.7 Novel Feature Engineering Framework (TCN model) - 12 -](#_Toc201927135)

[5.7.1 Temporal Sequence Features - 13 -](#_Toc201927136)

[5.7.2 Hierarchical Weekly Aggregates - 13 -](#_Toc201927137)

[5.7.3 Label Construction - 13 -](#_Toc201927138)

[6. Algorithmic Framework and Model Optimization - 14 -](#_Toc201927139)

[6.1 LightGBM - 14 -](#_Toc201927140)

[6.1.1 LightGBM Architectural Framework - 14 -](#_Toc201927141)

[6.1.2 Leaf-wise Tree Growth Strategy - 14 -](#_Toc201927142)

[6.1.3 Histogram-based Optimization - 15 -](#_Toc201927143)

[6.1.4 Regularization Mechanisms - 15 -](#_Toc201927144)

[6.1.5 Training Strategy - 15 -](#_Toc201927145)

[6.1.6 Limitations of LightGBM in Temporal Data Modeling - 15 -](#_Toc201927146)

[6.2 TCN - 16 -](#_Toc201927147)

[6.2.1 TCN Architectural Framework - 16 -](#_Toc201927148)

[6.2.2 Loss Function - 17 -](#_Toc201927149)

[6.2.3 Training Strategy - 17 -](#_Toc201927150)

[7. Performance - 17 -](#_Toc201927151)

[7.1 LightGBM - 17 -](#_Toc201927152)

[7.1.1 Week Data Model - 17 -](#_Toc201927153)

[7.1.2 Day Data Model Performance - 19 -](#_Toc201927154)

[7.2 TCN - 20 -](#_Toc201927155)

[7.2.1 Experimental Results - 20 -](#_Toc201927156)

[7.2.2 Ablation Study - 21 -](#_Toc201927157)

[8. Visualization - 24 -](#_Toc201927158)

[8.1 Confidence-accuracy Curve - 24 -](#_Toc201927159)

[8.2 Predicted Probability Distribution - 25 -](#_Toc201927160)

[8.3 ROC-AUC Curve - 26 -](#_Toc201927161)

[8.4 Average AUC Training Curve - 27 -](#_Toc201927162)

[8.5 Binary-logloss Training Curve - 27 -](#_Toc201927163)

[9. Reference - 28 -](#_Toc201927164)

# Introduction

Sports training injury risk analysis aims to predict the probability of sports injuries in advance by monitoring and modeling the training load and physiological state of athletes, helping coaches and athletes to develop scientific training plans, reduce the risk of sports injuries, and improve sports performance and athlete health. With the development of big data and machine learning technology, risk prediction methods based on historical training data have gradually become an important means of preventing sports injuries.

In this study, we designed and compared two main algorithms for sports injury risk prediction. The first one is the LightGBM model, which takes the sliding window aggregated statistics of training load indicators in multiple time windows (7 days, 14 days, and 28 days) as input, including the average, standard deviation, maximum and minimum values ​​in each window period. At the same time, the historical injury rate and the original training features are combined as auxiliary information. It should be noted that the calculation of the historical injury rate is strictly based on the historical records before the current sample window, and does not contain any label information from the present or the future, ensuring that there is no label leakage during the training process. The model output is a binary classification result of whether the athlete will suffer sports injuries in the next day, aiming to capture the impact of training load changes on injury risk at different time scales. The second model is the time series convolutional network (TCN) model. The TCN model uses the 5-dimensional time series features of the athlete for the past 28 consecutive days as input to capture time dependence and dynamic change trends. The model extracts time series features through a time series convolution structure, and finally outputs the probability of sports injuries on the 29th day, realizing a single-step future risk prediction based on continuous historical data.

The two methods respectively build models from the perspectives of statistical aggregation features and original time series features, combining the characteristics of sports training data, aiming to improve the accuracy and practicality of sports injury risk prediction.

# 2. Dataset

## 2.1 Data Source

The dataset originates from a Dutch high-level running team, spanning a 7-year period from 2012 to 2019. It focuses on middle- and long-distance runners competing in events ranging from 800 meters to the marathon. The initial dataset included 77 runners (27 female, 50 male) with an average team membership of 3.7 years, where most athletes competed at the national level and some at the international level. Data collection was standardized under a consistent head coach, ensuring uniformity in training regimes and data recording practices.

## 2.2 Event Distribution

After filtering out missing/anomalous data and consecutive injuries within 3 weeks, the dataset retained 74 athletes. The event distribution is as follows:

* **Day:** 42,183 healthy events and 583 injury events.
* **Week:** 42,223 healthy events and 575 injury events.

Injury events account for approximately 1.4% of total records, indicating a severe class imbalance between healthy and injury cases.

## 2.3 Collection Methods

* **Objective Data**: Recorded via GPS watches with heart rate monitors, including: Running distance, duration, and heart rate zones (Z1–Z5, where Z5 represents near-maximum heart rate).
* **Subjective Data**: Athlete self-reports on; Perceived exertion (post-training tiredness), training success (session quality), and recovery (pre-training restfulness).
* **Injury Identification**: Injuries are flagged in training logs as "unable to complete sessions due to injury," with a 3-week injury-free period required prior to the event for valid injury labeling.

# 3. Feature Description

## 3.1 Day Type

The day approach constructs 70 features from 7 days of training data preceding an event, with 10 features per day:

Table 1. Daily activity metrics.

|  |  |
| --- | --- |
| **nr. sessions** | Number of daily training sessions. |
| **total km** | Daily running distance (km). |
| **km Z3-4** | Distance (km) in intensity zones 3–4 (anaerobic threshold or above). |
| **km Z5-T1-T2** | Distance (km) in zone 5 (near max heart rate) or track intervals (T1 = long reps, T2 = short reps). |
| **km sprinting** | Sprint training distance (km). |

Table 2. Training types.

|  |  |
| --- | --- |
| **strength training** | Binary indicator (1/0) for strength training on the day. |
| **hours alternative** | Cross-training hours (e.g., cycling, swimming). |

Table 3. Subjective ratings.

|  |  |
| --- | --- |
| **perceived exertion** | Post-training tiredness (0–1, -0.01 for rest days). |
| **perceived trainingSuccess** | Training quality rating (0–1, -0.01 for rest days). |
| **perceived recovery** | Pre-training restfulness (0–1, -0.01 for rest days). |

Suffixes denote days before the event (e.g., nr. sessions.6 = day before event, nr. sessions.0 = 7 days before).

## 3.2 Week Type

The week approach uses 69 features from 3 weeks of aggregated training data (22 weekly features + 3 relative distance change metrics):

Table 4. Weekly activity summaries.

|  |  |
| --- | --- |
| **nr. sessions** | Total weekly training sessions. |
| **nr. rest days** | Weekly rest days without training. |
| **total kms** | Weekly running mileage (km). |
| **max km one day** | Maximum daily running distance in the week. |

Table 5. Intensity metrics.

|  |  |
| --- | --- |
| **total km Z3-Z4-Z5-T1-T2** | Total distance (km) in zones 3–5 or track intervals. |
| **nr. tough sessions** | Sessions in zone 5, T1, or T2 (max effort/intensive intervals). |
| **total / max km Z3-4 one day** | Weekly distance and daily max in zones 3–4. |
| **total / max km Z5-T1-T2 one day** | Weekly distance and daily max in zone 5 or track intervals. |

Table 6. Cross-training & strength.

|  |  |
| --- | --- |
| **total hours alternative training** | Weekly cross-training hours. |
| **nr. strength trainings** | Weekly strength sessions. |

Table 7. Aggregated subjective ratings.

|  |  |
| --- | --- |
| **avg/min/max exertion** | Weekly average, minimum, and maximum post-training tiredness. (0-1) |
| **avg/min/max training success** | Weekly average, minimum, and maximum training quality ratings. (0-1) |
| **avg/min/max recovery** | Weekly average, minimum, and maximum pre-training restfulness. (0-1) |

Suffixes denote weeks before the event (e.g., nr. sessions = week before event, nr. sessions.1 = 2 weeks before, nr. sessions.2 = 3 weeks before).

## 3.3 Common Dataset Structure

Table 8. Common dataset structure.

|  |  |
| --- | --- |
| **Binary Label** | injury (1 = injury, 0 = healthy). |
| **Athlete ID** | Anonymized identifier for individual athletes. |
| **Date** | Event day relative to the first record (converted to datetime format in preprocessing). |

## 3.4 Key Considerations

The dataset's primary challenges include severe class imbalance (1.4% injury rate) and the need to capture temporal dependencies in training loads. The day approach emphasizes short-term training patterns (7 days), while the week approach focuses on long-term trends (3 weeks), enabling complementary insights into injury risk factors.

# 4. Data processing

## 4.1 Feature Name Standardization

The preprocessing pipeline initiates with the standardization of feature nomenclature to ensure syntactic uniformity. This process involves the removal of all parenthetical notations and punctuation from column headers, addressing potential inconsistencies that could impede model training. For instance, features such as km (Z3-4) are normalized to km Z3-4, establishing a consistent naming convention that facilitates seamless integration with machine learning frameworks.

## 4.2 Temporal Data Conversion and Sorting

Raw date indices are converted to datetime objects by offsetting from a fixed reference point (e.g., 2019-01-01), enabling rigorous temporal analysis. Data records are subsequently sorted by athlete identifier and chronological order to maintain the temporal integrity of training logs, a critical step for preserving the causal relationship between training loads and injury occurrences.

## 4.3 Feature Filtering and Imputation

Low-variance features (with variance < 0.001) are pruned to reduce noise, while missing values are imputed using median values. This dual-step process ensures feature relevance and data completeness, enhancing model generalization by eliminating non-informative attributes and resolving data scarcity.

## 4.4 Challenge of Class Imbalance

The dataset exhibits a severe class imbalance, with injury events comprising only 1.4% of total observations. This imbalance poses significant challenges for machine learning models, which tend to be biased toward the majority (healthy) class, leading to poor recall of rare injury cases. To address this, three resampling strategies were implemented, focusing on synthetic over-sampling and hybrid techniques to maintain data integrity while balancing class distributions.

## 4.5 SMOTE: Synthetic Minority Over-sampling Technique

### 4.5.1 Principle and Implementation

SMOTE is a leading over-sampling method for imbalanced datasets, operating on the following principles:

* For each minority class (injury) sample, SMOTE identifies its k-nearest neighbors (typically k=5) in the feature space.
* New synthetic samples are created by interpolating between the minority sample and its neighbors, preserving the local structure of the data.
* By generating samples within the existing minority class manifold, SMOTE avoids introducing artificial patterns while increasing minority class representation.

### 4.5.2 Advantages for Injury Prediction

* Unlike random over-sampling, SMOTE generates semantically meaningful samples that reflect real-world training-injury relationships.
* Synthetic samples retain the statistical dependencies between training load metrics, ensuring the model learns genuine injury risk patterns.

## 4.6 SMOTE-ENN: Hybrid Over-sampling and Under-sampling

### 4.6.1 Overview

SMOTE-ENN is a hybrid strategy combining SMOTE with the Edited Nearest Neighbors (ENN) under-sampling technique:

* **SMOTE Over-sampling**: First applies SMOTE to increase minority class samples, as described above.
* **ENN Cleaning**: Removes borderline samples that are misclassified by their k-nearest neighbors, using the following steps:

1. For each sample, identify its k-nearest neighbors (k=3).
2. Remove samples whose class differs from the majority of their neighbors, reducing noise and improving class separability.

### 4.6.2 Synergistic Benefits

* ENN mitigates the risk of SMOTE generating ambiguous samples near class boundaries.
* By cleaning the dataset after over-sampling, SMOTE-ENN reduces variance and enhances the model's ability to generalize to unseen data.

## 4.7 Data Processing Pipeline

### 4.7.1 Feature Integration

Before resampling, the two feature spaces (temporal sequences and weekly aggregates) are concatenated to form a unified feature matrix:

* Temporal sequence features .
* Weekly aggregate features .
* Concatenated matrix  integrates both short-term dynamics and long-term context,

where is the number of samples, is the number of time steps in the temporal window, is the number of features per time step, and is the dimensionality of the weekly aggregate features.

### 4.7.2 Resampling Workflow

* **SMOTE-based Over-sampling:** For each injury sample, k-nearest neighbors are identified in the combined feature space. Synthetic injury samples are generated along feature space trajectories between existing injury cases, increasing the minority class size to balance the dataset.
* **SMOTE-ENN Hybrid Processing:** Following SMOTE, the ENN algorithm evaluates each sample's neighborhood consistency. Inconsistent samples (misclassified by their neighbors) are removed, refining the dataset for clearer class boundaries.
* **Data Shuffling:** Resampled data is randomly shuffled to break temporal dependencies, ensuring model training is not biased toward sequential patterns.

### 4.7.3 Methodological Advantages

* By operating on combined features, the resampling strategies preserve the relationship between fine-grained temporal dynamics and macrolevel training context.
* SMOTE-ENN dynamically adjusts the class ratio, avoiding overfitting caused by excessive over-sampling while addressing imbalance.
* Resampled datasets enable models to learn injury risk patterns without being dominated by the majority class, leading to more reliable risk predictions.

# 5. Feature Engineering for Temporal Pattern Recognition

## 5.1 Sliding Window Time-Series Metrics (LightGBM model)

LightGBM model uses sliding window features 7/14/28 days before the event and original events as input data, and the prediction target is the damage state (y = 1 if injury happen, and y = 1 if injury not happen) on the event day.

Chronological training data is transformed using sliding windows (7, 14, 28 days) to derive dynamic features, including:

* **Aggregate Statistics**: Mean, standard deviation, maximum, and minimum values for training load metrics (e.g., distance, intensity zone kilometers),
* **Injury Recurrence Rate**: Proportion of injury events within each window, capturing short-term and long-term risk trends.

## 5.2 Feature Space Transformation

The engineering pipeline constructs a rich feature space by:

* Summarizing daily training metrics into weekly aggregates for macrolevel analysis.
* Separating running distances by heart rate zones (Z3–Z5, T1–T2) to distinguish between aerobic and anaerobic training loads.
* Combining GPS-derived objective data (e.g., distance, duration) with athlete self-reports (perceived exertion, recovery) to capture holistic training stress.

## 5.3 Feature Importance Hierarchy

Engineered features are designed to reflect:

* Short-term fluctuations in intensity (e.g., km Z5-T1-T2\_mean\_7d).
* Long-term cumulative load (e.g., total kms\_std\_28d).
* Individualized z-score normalization using healthy event statistics, accounting for inter-athlete variability in training capacity.

## 5.4 Temporal direction constraints

* Ensure that the rolling window only contains historical data by setting the **closed=’left’** parameter, blocking future information from flowing into feature calculations at the algorithmic level.
* Use **TimeSeriesSplit** to divide the training set and test set, forcing the validation set's time window to be completely after the training set.

The two method eliminates the possibility of feature leakage, and the feature can be considered as one of the factors in injury prediction.

## 5.5 Modeling of the cumulative effect of damage risk and explanation of rationality

The historical risk exposure level is quantified by calculating the occurrence frequency of injury events within the window period (injury rate). This feature design is based on the following biological mechanisms:

* **Training load-recovery imbalance**

The 7-14 days after injury is a critical period for tissue remodeling. At this time, an ACWR (acute/chronic load ratio) > 1.5 will significantly increase the risk of re-injury [7].

* **Cumulative fatigue effect**

Within 28 days after injury, athletes' performance significantly declines (vertical jump height reduction > 16%), accompanied by an increase in anxiety occurrence rate (> 43%), forming a dual physiological and psychological stress source, which increases the risk of re-injury to 3.4 times the baseline level [8].

## 5.6 Limitations of Original Sliding Window Features (LightGBM model)

The initial feature engineering approach, relying on sliding window statistics (mean, std, max, min) over 7, 14, 28-day windows, presented critical limitations for injury prediction:

* **Loss of Temporal Pattern Recognition**: Aggregating features into static statistics (e.g., total km\_mean\_7d) discarded sequential dependencies in training loads, failing to capture dynamic patterns like sudden intensity increases or recovery cycles that precede injuries.
* **High Dimensionality and Noise**: The resulting feature space (e.g., 70 features for day-level analysis) included redundant metrics (e.g., mean/std of similar intensity zones), increasing model complexity without proportional gain in predictive power.
* **Inadequate Hierarchical Representation**: The approach treated short-term (7-day) and long-term (28-day) features uniformly, neglecting the biological reality that overuse injuries often emerge from cumulative chronic strain combined with acute load spikes.
* **Suboptimal Feature Selection**: Including all numeric features without domain-driven prioritization led to dilution of critical signals (e.g., high-intensity training volume) among less relevant metrics.

## 5.7 Novel Feature Engineering Framework (TCN model)

To address these limitations, a new feature engineering pipeline was developed to construct two complementary feature spaces:

* Temporal sequence features for convolutional modeling and hierarchical weekly aggregates for contextual understanding.
* The **input data** of TCN model is the 5-dimensional time series feature of the past 28 consecutive days, and the final output is the probability of whether the sports injury occurs on the 29th day.

### 5.7.1 Temporal Sequence Features

This component captures fine-grained training dynamics over a 28-day input window, focusing on five clinically relevant metrics:

* Cumulative daily mileage as a proxy for overall training volume.
* Daily distance in anaerobic threshold zones (Z3–Z4), a key indicator of physiological stress.
* Binary indicator of daily strength sessions, reflecting recovery and injury prevention efforts.
* Athlete-reported post-training fatigue, capturing subjective physiological response.
* Athlete-reported pre-training restfulness, indicating readiness to train.

By preserving the temporal order of these features, the sequence matrix enables models like TCNs to learn:

* Sudden increases in high-intensity training within short windows.
* Correlations between poor perceived recovery and subsequent injury risk.
* Streaks of consecutive high-load days without adequate rest.

### 5.7.2 Hierarchical Weekly Aggregates

Complementing the sequence features, weekly aggregates over a 4-week window (28 days) provide macrolevel training context, structured into six semantically meaningful metrics per week:

* **Training Frequency**: Number of sessions in the week, reflecting training consistency.
* **Strength Training Compliance**: Number of rest days without strength training, indicating potential muscular imbalance risks.
* **Weekly Mileage**: Total weekly running distance, capturing chronic training load.
* **High-Intensity Volume**: Total weekly distance in Z3–Z4 zones, relating to cumulative physiological strain.
* **Average Exertion**: Mean perceived exertion across the week, reflecting training intensity perception.
* **Minimum Recovery**: Worst perceived recovery in the week, indicating periods of inadequate rest.

This hierarchical structure  enables models to learn:

* Imbalances between long-term training capacity and short-term load spikes.
* Training cycle patterns (e.g., tapering or overload phases) and their impact on injury risk.
* Individual tolerance thresholds for weekly high-intensity volume.

### 5.7.3 Label Construction

Labels are defined as binary indicators of injury occurrence within a 7-day forecast window following the input window:

* if any injury event occurs in the forecast window,
* for injury-free forecast periods,

where denotes the label “injury”.

This definition aligns with practical injury prevention needs, enabling proactive intervention 1–7 days before potential injury onset.

# 6. Algorithmic Framework and Model Optimization

## 6.1 LightGBM

LightGBM (Light Gradient Boosting Machine) is a state-of-the-art gradient boosting decision tree framework optimized for high-dimensional classification tasks. It constructs an ensemble of decision trees sequentially, where each tree corrects the errors of its predecessor, enabling the capture of complex non-linear relationships between training load metrics and injury outcomes. This design addresses the dataset's key challenges: high feature dimensionality (69/70 features), severe class imbalance (1.4% injury rate), and the need to model temporal dependencies.

### 6.1.1 LightGBM Architectural Framework

The following figure illustrates the framework of training model with LightGBM

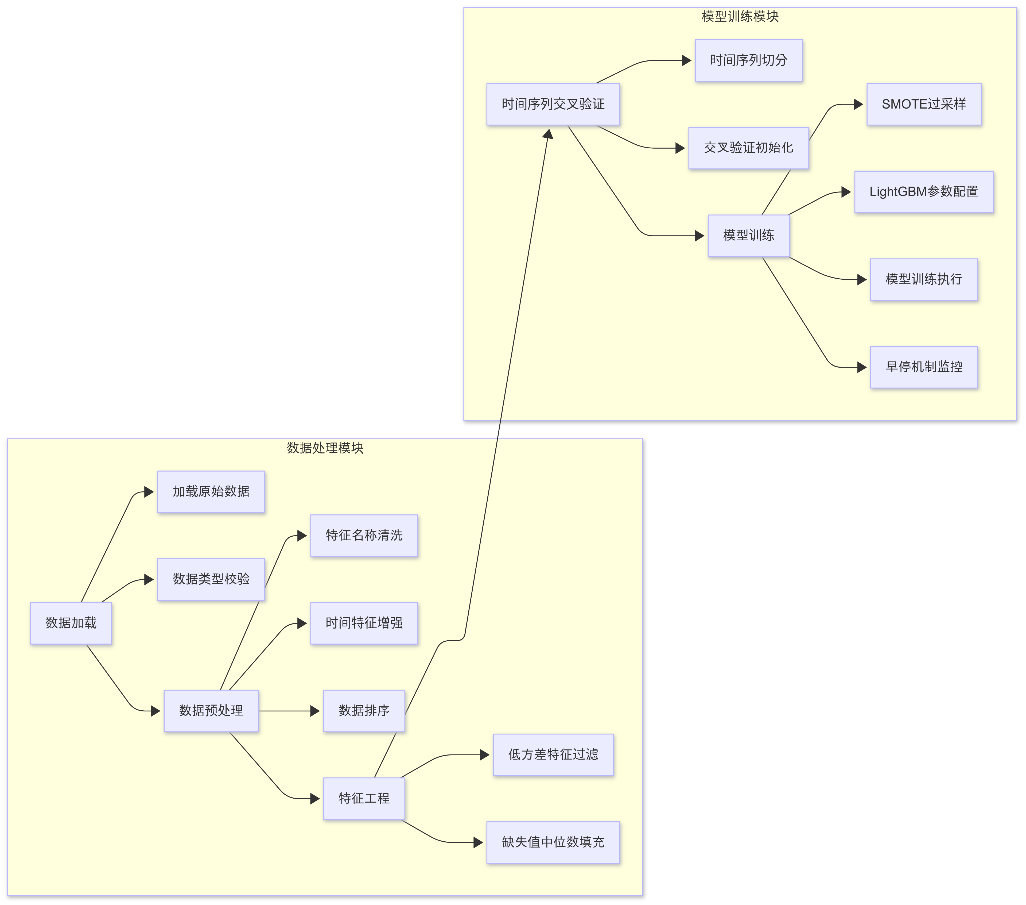


Figure 1. The core architecture of LightGBM.

### 6.1.2 Leaf-wise Tree Growth Strategy

LightGBM differentiates from traditional GBDT frameworks via a leaf-wise (best-first) growth strategy:

* Expands the leaf with the highest information gain, reducing node count for complex pattern capture.
* Shallow trees prevent overfitting while maintaining predictive power.
* Minimizes splits for faster training and lower memory consumption.

### 6.1.3 Histogram-based Optimization

The algorithm accelerates training through histogram-based discretization.

* Discretizes continuous features (e.g., total km) into intervals, reducing unique values.
* Processes binned features to reduce split complexity from  to  ( is bin count, typically 255).

### 6.1.4 Regularization Mechanisms

* **L1/L2 Regularization**:  and  penalize complex tree structures, preventing overly confident predictions.
* **Min Child Samples**: Enforces a minimum number of data points in each leaf, ensuring that splits are statistically meaningful and reducing the risk of overfitting on small fluctuations.
* **Feature Subsampling**: By randomly selecting a subset of features for each iteration, this technique introduces noise and decorrelates trees, acting as a form of implicit regularization.

### 6.1.5 Training Strategy

* **Class Imbalance Handling**

1. Generates synthetic injury samples to balance the training set to 50% injury class.
2. Upweights injury cases during loss calculation.

* **Hyperparameter Configuration**

1. Num Leaves=31.
2. Learning Rate=0.1.
3. n\_estimators=300. Forms a robust ensemble with early stopping.
4. Force Col-wise

* **Time-series Cross-Validation**

1. 5-fold TimeSeriesSplit (TSS).
2. Halts training if validation AUC does not improve for 50 iterations.
3. Chooses the model with highest validation AUC.

### 6.1.6 Limitations of LightGBM in Temporal Data Modeling

* Relies on pre-engineered features (e.g., sliding windows) rather than explicit long-range dependency modeling, hindering learning of time-lagged relationships like cumulative load effects.
* Treats training events independently post-aggregation, failing to model causal sequences (e.g., high-intensity training followed by inadequate recovery).
* Complex tree growth for fine-grained time series (7–28-day windows) risks overfitting and reduces training speed.

## 6.2 TCN

In order to more effectively capture the impact of training load changes over time on sports injuries, we used a temporal convolutional network (TCN) to replace LightGBM. In contrast, TCN has significant advantages in time series modeling capabilities, dynamic risk identification, and training efficiency:

* TCN uses a dilated convolution mechanism to extract contextual features at exponential intervals (such as 1, 2, 4, and 8 days) in the time dimension, thereby capturing both short-term (such as 7-day acute load) and medium-term (such as 28-day chronic strain) trends, which is highly consistent with the importance of the "acute-slow load ratio" in injury prediction in practice.
* In order to ensure that the model's prediction does not rely on future information, TCN adopts a causal convolution structure and removes invalid outputs at the end of the time step through the Chomp operation, effectively simulating the task setting of "predicting the future only using known history" in reality.
* Unlike LightGBM, TCN can retain the temporal order of training data and learn complex cross-time step interactions.
* We further combine Focal Loss to deal with the problem of sample imbalance in the data where the minority class is "injured".

### 6.2.1 TCN Architectural Framework

The TCN consists of input, TemporalConvNet, which is a cascade of three Temporal Blocks, linear layer and output.

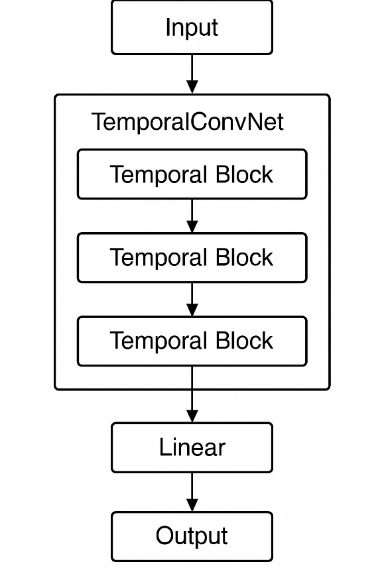


Figure 2. The core architecture of TCN.

The following is a detailed architecture of the Temporal Block: it contains convolutional layers, causal convolutional layers, ReLU activation functions, dropout layers, and downsampling layers.

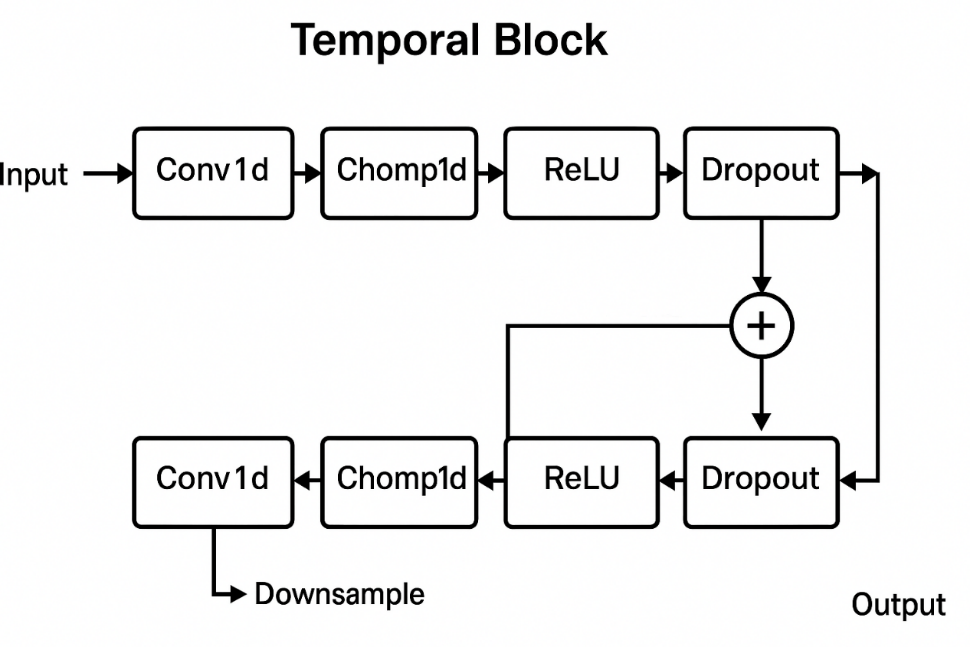


Figure 3. The detailed implementation of Temporal Block.

### 6.2.2 Loss Function

As the 1.4% injury rate, we use Focal loss for dealing with the imbalance.

where for down-weighting healthy class and for reducing weight of easy examples.

### 6.2.3 Training Strategy

* Using dropout=0.2 to prevent overfitting in temporal blocks.
* Halts training after 50 epochs without validation gain.
* Using normal distribution (*μ*=0, *σ*=0.01) for weight initialization, to make sure stable convergence.

# 7. Performance

## 7.1 LightGBM

### 7.1.1 Week Data Model

The optimal threshold is determined as **0.3535** to meet the recall target. This threshold is derived by maximizing Youden's Index, balancing the true positive rate and false negative rate.

Table 9. Validation set performance.

|  |  |  |
| --- | --- | --- |
| Metric | Value | Explanation |
| AUC | 0.7762 | The model shows strong ability to distinguish between healthy and injured samples, surpassing the random level (0.5). |
| F1 Score | 0.0520 | The harmonic mean of precision and recall is low, significantly affected by class imbalance (1.4% injury rate). |
| Precision | 0.0269 | Only 2.69% of samples predicted as injured are actual injuries, indicating extremely high false positive rates. |
| Accuracy | 0.6241 | Moderate overall accuracy, dominated by the majority (healthy) class. |
| Specificity | 0.6221 | Average ability to correctly identify healthy samples, with 27.79% of healthy samples misclassified as injured. |
| Recall | 0.7674 | Successfully identifies 76.74% of injury cases, demonstrating the model's capability to capture minority classes. |

Table 10. Test set performance.

|  |  |  |
| --- | --- | --- |
| Metric | Value | Explanation |
| AUC | 0.7234 | 5.38% decrease from the validation set, reflecting limited generalization ability on unseen data. |
| F1 Score | 0.0460 | Further performance decline, suggesting potential unbalanced class distribution in the test set. |
| Precision | 0.0238 | Reduced prediction accuracy, exacerbating false positive issues. |
| Accuracy | 0.6217 | Slightly consistent with the validation set, dominated by majority class predictions. |
| Specificity | 0.6210 | Slight decrease in healthy sample identification, indicating model stability issues. |
| Recall | 0.6783 | 8.93% decrease in injury identification, suggesting weak adaptability to new data. |

Table 11. Confusion matrix performance.

|  |  |  |
| --- | --- | --- |
|  | Validation Set | Test Set |
| **TN** (true negative) | 21014 | 5244 |
| **FP** (false positive) | 12764 | 3201 |
| **FN** (false negative) | 107 | 37 |
| **TP** (true positive) | 353 | 78 |

**Core Issue**: The number of FPs is 36 times that of TPs, reflecting severe over-prediction of the minority class, directly related to the 1.4% injury rate in the dataset.

### 7.1.2 Day Data Model Performance

The optimal threshold is **0.1515**, lowering the threshold to enhance recall and adapt to the practical need for "early detection and intervention."

Table 12. Validation set performance.

|  |  |  |
| --- | --- | --- |
| Metric | Value | Explanation |
| AUC | 0.8680 | Significantly better discrimination ability than the weekly model, indicating daily-scale features are more sensitive for injury prediction. |
| F1 Score | 0.0981 | 88.65% improvement over weekly data, but still low, highlighting persistent class imbalance issues. |
| Precision | 0.0525 | 1.95 times higher than the weekly model, indicating slightly better prediction reliability for daily data. |
| Accuracy | 0.8114 | Higher overall accuracy, with a clear advantage in healthy class prediction. |
| Specificity | 0.8122 | Strong ability to identify healthy samples, with only 18.78% misclassified as injured. |
| Recall | 0.7532 | Close to the 75% recall target, with injury capture capability comparable to the weekly model. |

Table 13. Test set performance.

|  |  |  |
| --- | --- | --- |
| Metric | Value | Explanation |
| AUC | 0.7443 | 14.25% decrease from the validation set, with more severe overfitting than the weekly model. |
| F1 Score | 0.0775 | 21.00% decrease from the validation set, indicating weaker generalization than weekly data. |
| Precision | 0.0415 | Reduced precision, reigniting false positive issues. |
| Accuracy | 0.8080 | Slightly consistent with the validation set, dominated by majority class predictions. |
| Specificity | 0.8111 | Stable healthy sample identification, demonstrating sufficient learning of the majority class. |
| Recall | 0.5897 | 21.70% significant decrease in injury identification, suggesting poor generalization of temporal features in daily data. |

Table 14. Confusion matrix performance.

|  |  |  |
| --- | --- | --- |
|  | Validation Set | Test Set |
| **TN** (true negative) | 27409 | 6843 |
| **FP** (false positive) | 6337 | 1594 |
| **FN** (false negative) | 115 | 48 |
| **TP** (true positive) | 351 | 69 |

**Improvement**: FP count reduced by 50.13% compared to weekly data, but FN slightly increased, reflecting a trade-off imbalance between "reducing false positives" and "enhancing recall."

## 7.2 TCN

### 7.2.1 Experimental Results

In our proposed model, we perform temporal modeling on five selected features to predict the risk of sports-related injuries. The dataset is first loaded and processed, followed by feature construction. We then split the data into training, validation, and test sets in a 60%, 20%, and 20% ratio, respectively.

To address the severe class imbalance in the original dataset, we apply the SMOTE + ENN strategy to the training set. This approach generates synthetic minority samples while simultaneously removing noisy or ambiguous instances, effectively balancing the positive and negative sample distribution within the training set. Importantly, the validation and test sets remain untouched and reflect the original data distribution. By doing so, we ensure that the model is trained on a balanced dataset while maintaining unbiased evaluation through the original validation and test data. This design allows the validation set to serve as a reliable reference for adjusting and correcting the training process, even when the training data distribution is modified.

Based on this configuration, we conducted a series of experiments to assess the model's predictive performance and compare it with the baseline LightGBM model.

Table 15. Model performance comparison.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1 | AUC |
| LightGBM | 0.0415 | 0.5897 | 0.0775 | 0.7443 |
| TCN | 0.9697 | 0.9057 | 0.9366 | 0.9997 |

The results demonstrate a significant improvement in predictive performance with the application of the TCN model. Compared to the baseline LightGBM model, which achieved a precision of only 0.0415, the TCN model attained a much higher precision of 0.9697. This indicates a substantial reduction in false positive predictions, greatly improving the model’s reliability in real-world scenarios.

In terms of recall, the TCN model also outperformed LightGBM by a large margin (0.9057 vs. 0.5897), suggesting that it was able to correctly identify a much greater proportion of injury cases. The F1 score and AUC further highlight the comprehensive performance gain: the TCN model achieved an F1 score of 0.9366 and an AUC of 0.9997, compared to 0.0775 and 0.7443 for LightGBM, respectively.

These results indicate that the TCN model can effectively capture temporal patterns and dependencies in the training data, allowing it to distinguish between injury and non-injury cases with near-perfect accuracy. Overall, the application of TCN significantly enhances the model’s ability to detect injury risks while minimizing false alarms, achieving superior performance across all evaluation metrics.

### 7.2.2 Ablation Study

To further validate the effectiveness of our selected modules and feature design, we conducted ablation study. First, we focus on the impact of the SMOTE + ENN sampling strategy on model performance.

We first examined the direct changes in sample distribution before and after applying SMOTE+ENN. Without using SMOTE+ENN, the training set was highly imbalanced, containing 366 positive samples and 25,272 negative samples, resulting in a sample weight of approximately **69.05**. In contrast, after applying SMOTE+ENN, the training set became much more balanced, with 25,272 positive and 21,799 negative samples, and the corresponding sample weight dropped significantly to **0.86**. This reflects a substantial improvement in sample balance, which is expected to positively influence model learning.

To analyze the performance impact, we trained the model for epochs and recorded metrics at epoch 10, 15, and 20. The results are summarized as follows:

Table 16. Ablation results of SMOTE + ENN module.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Epochs | ╳ SMOTE+ENN | √ SMOTE+ENN |
| Loss | 10 | 0.0011 | 0.0009 |
| 15 | 0.0029 | 0.0005 |
| 20 | 0.0007 | 0.0003 |
| Precision | 10 | 0.9839 | 0.9412 |
| 15 | 0.4435 | 0.8750 |
| 20 | 0.7778 | 0.9697 |
| Recall | 10 | 0.5755 | 0.7547 |
| 15 | 0.9623 | 0.9245 |
| 20 | 0.9906 | 0.9057 |
| F1 | 10 | 0.7262 | 0.8377 |
| 15 | 0.6071 | 0.8991 |
| 20 | 0.8714 | 0.9366 |
| AUC | 10 | 0.9915 | 0.9965 |
| 15 | 0.9970 | 0.9990 |
| 20 | 0.9996 | 0.9997 |

In contrast, after using SMOTE + ENN, Recall is significantly improved, the minority class is easier to identify, Precision and Recall are better balanced, F1 score is improved, AUC is close to 1, and the model is more robust and has stronger generalization ability.

These results confirm that SMOTE + ENN plays a key role in reducing class imbalance. When SMOTE + ENN is not used, the proportion of injury samples in the training set is very low, and the model is prone to "bias towards the majority class". We use SMOTE to synthesize new samples, and ENN removes inconsistent noise samples, making the model learning process more sensitive to minority classes. The final result Recall is significantly improved, for example, from 0.5755 to 0.7547 at epoch=10, indicating that the model's recognition rate of injury samples has been greatly improved. The original model has an imbalance problem between Precision and Recall (such as Precision = 0.4435 at epoch=15). After adding SMOTE + ENN, Precision and Recall increased evenly, and the F1 score improved across the board. When epoch=20, F1 went from 0.8714 to 0.9366, and when epoch=15, F1 went from 0.6071 to 0.8991, indicating that the model is more stable and will not just "blindly detect" or "be too conservative." After adding SMOTE + ENN, the model AUC approaches the ideal value, and the generalization ability is enhanced.

In order to further evaluate the specific contribution of each input feature to the model performance, we conducted a systematic ablation experiment on the original features. The original input includes five core features, namely:

* Daily training mileage (total km).
* High-intensity training mileage (km Z3-4).
* Whether to conduct strength training (strength training).
* Self-assessment of fatigue level after training on the same day (perceived exertion).
* Self-assessment of recovery status before training (perceived recovery).

In the ablation experiment, we use masking to set the input of each feature position to zero, and keep the other features unchanged to observe the changes in model performance. Through this single-factor intervention design, we can quantitatively evaluate the strength and importance of each feature in the overall prediction task, thereby providing a basis for subsequent feature optimization and simplification. The relevant experimental results are as follows:

Table 17. Feature ablation results.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Loss | Precision | Recall | F1 | AUC |
| ① | 0.0003 | 0.9697 | 0.9057 | 0.9366 | 0.9997 |
| ② | 0.0003 | 1.0000 | 0.8679 | 0.9293 | 1.0000 |
| ③ | 0.0005 | 0.8814 | 0.9811 | 0.9286 | 0.9998 |
| ④ | 0.0009 | 0.7391 | 0.9623 | 0.8361 | 0.9993 |
| ⑤ | 0.0007 | 0.9263 | 0.8302 | 0.8756 | 0.9972 |
| ⑥ | 0.0004 | 1.0000 | 0.8774 | 0.9347 | 0.9998 |

The first group was used as the control group, using all five original features. In the second to sixth groups, each feature was masked separately, that is, the corresponding feature was set to zero, and the rest remained unchanged, in order to observe the changes in model performance after each feature was masked. The masking order of the five input features is: total km, km Z3-4, strength training, perceived exertion, perceived recovery.

From the results, it can be seen that the best overall performance is the complete feature group (group 1), which has the most balanced performance in F1 value and AUC. After masking "whether to do strength training" (group 4), the F1 value dropped most significantly (dropped to 0.8361), indicating that this feature has a key impact on the model's judgment, especially in accurately distinguishing positive and negative samples. After masking "daily training mileage" (group 2) and "subjective recovery status" (group 6), the model precision increased to 1.0, but the recall decreased to a certain extent, indicating that the model has become more conservative. Although there is almost no false alarm, it may miss some injury risks. Masking "high-intensity training mileage" (group 3) has the least impact on recall, and it has increased slightly, indicating that this feature has some redundancy in capturing injuries, but may have information overlap with other features. After masking "subjective fatigue" (group 5), both F1 and AUC decreased slightly, indicating that it has an auxiliary but non-core position in the overall judgment of the model.

# 8. Visualization

## 8.1 Confidence-accuracy Curve

The counterintuitive phenomenon where accuracy approaches 1.0 in the 0.25-0.5 confidence interval and drops to near 0 in the 0.5-0.75 interval is closely related to data distribution and model decision logic:

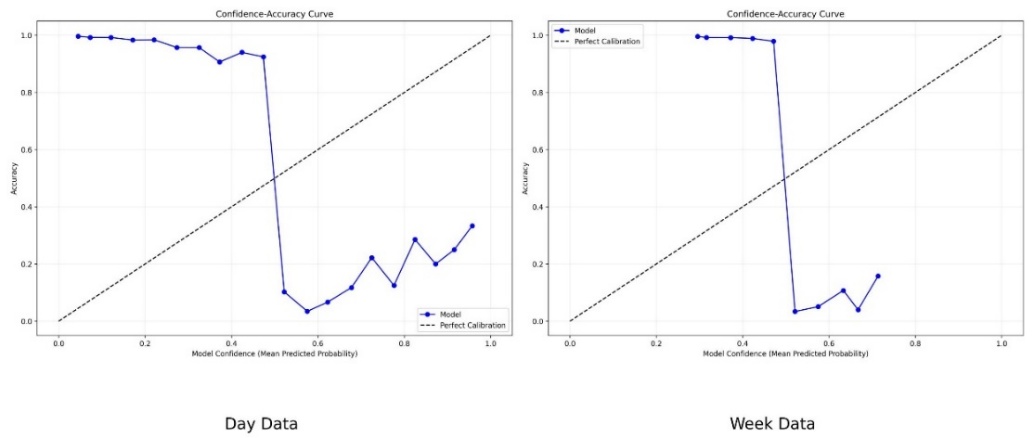


Figure 4. Confidence-accuracy curves of day (left) and week (right) data.

For day data:

* Curve features:

1. Low confidence interval (0-0.4): Accuracy is significantly higher than confidence (near 1.0), and the model predicts reliably for "unambiguously healthy samples."
2. Medium confidence interval (0.4-0.6): accuracy falls off a cliff below 0.1, and the model classifies "ambiguous examples" poorly.
3. High confidence range (0.6-1.0): volatility picks up but local shocks, extreme high confidence forecasts are still biased.

* Possible influencing factors:
  1. With only 1.4% of the samples being injured, the model naturally tends to predict the healthy class (low confidence). The low confidence samples are mostly boundary cases, and the actual labels are consistent with the predicted direction, reflecting the conservative tendency of the model for the majority class.
  2. The daily data rely on sliding window statistics, but the original features do not explicitly model the time dependence, which may be the traditional sliding window missing dynamic patterns. Medium-confidence samples may be in periods of abrupt changes in training load (e.g., after a short period of intense training), but the model fails to capture the temporal association, leading to misjudgment.
  3. The daily features contain repeated dimensions (such as the overlap between Z3-Z4 and Z5-T1-T2 intervals). The medium confidence intervals may correspond to samples with similar feature weights, and the model has difficulty distinguishing key signals (such as the conflict between subjective fatigue perceived <e:1> and objective distance total km).

For week data:

* Curve features:

1. Low confidence range (0-0.2): Accuracy is near perfect (1.0), and the model is extremely stable on long-term stable negative examples.
2. Medium confidence range (0.2-0.6): Large oscillations (accuracy drops to near 0 at 0.4), showing "high-confidence counterintuitive failure."
3. High confidence range (0.6-1.0): stable recovery (0.8-1.0 accuracy >0.8), reliable long-term trend prediction.

* Possible influencing factors:
  1. Weekly features smooth out daily level noise by aggregating features (e.g., total distance per week), but oversmoothing can cause key short-term variations to be lost (e.g., sudden increases in intensity training during the week go uncaptured), triggering medium-confidence misjudgment.
  2. Compared with the daily data, the weekly data has fewer injury samples, but the time window is longer, which includes 3 weeks, and the distribution of injury events is sparser. Oversampling of weekly data by SMOTE may introduce spurious samples, resulting in the accumulation of medium confidence interval noise.

## 8.2 Predicted Probability Distribution

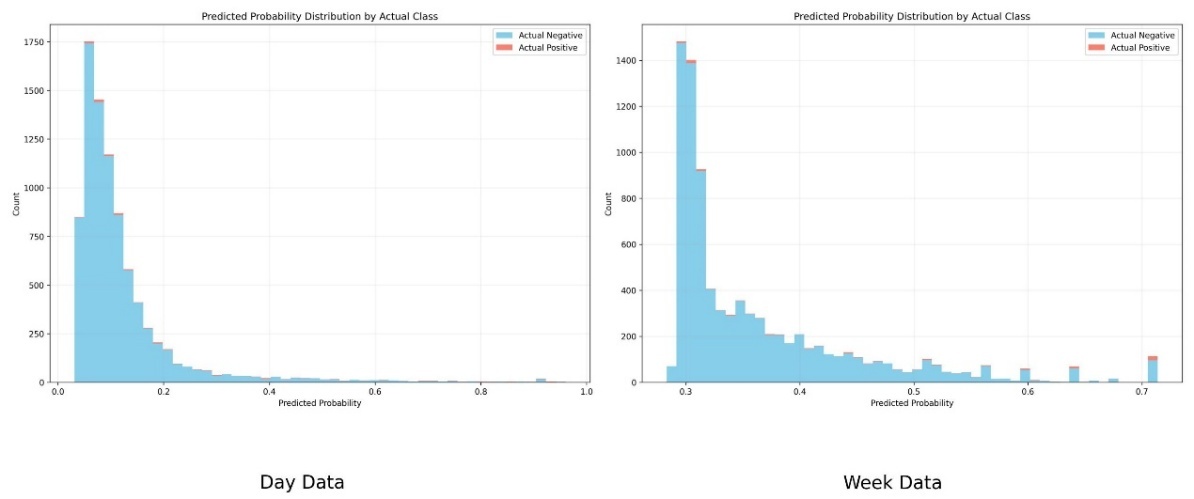


Figure 5. Predicted probability distribution of day (left) and week (right) data.

* Data distribution characteristics:

1. Healthy samples (negative class) have a count exceeding 1,400 in the 0-0.3 probability interval, accounting for over 70% of total samples, forming a dense "low-risk cluster." To adapt to the majority class distribution, the model suppresses most samples' probabilities below 0.3, making the low-probability interval a "safe zone" for healthy classes.
2. Injury samples (positive class) are sparsely distributed in the high-probability interval (0.6-0.7), with the highest prediction probability only 0.7. Due to the extremely low injury event rate (1.4%) in training data, LightGBM avoids overfitting to rare events through a "conservative strategy"—even if samples exhibit injury features, the model tends to give moderate probabilities (≤0.7) rather than high-risk judgments (>0.7).

* Causes of abnormal prediction range:
  1. Although synthetic injury samples are generated via SMOTE, new samples only interpolate within the feature space of existing positive classes, not breaking through the probability distribution boundary of original data. The model lacks generalization ability for "extreme injury features" not seen in training, resulting in prediction probabilities not exceeding 0.7.
  2. Gradient boosting tree models construct decision boundaries by splitting nodes, but injury samples have dispersed feature distributions (e.g., abnormal combinations of high-intensity training distance and recovery scores), making it difficult to form high-confidence decision paths. The model's output probability essentially represents "the proportion of positive classes in leaf nodes," and since the positive class proportion in leaf nodes is at most 70%, the maximum prediction probability is 0.7.

## 8.3 ROC-AUC Curve

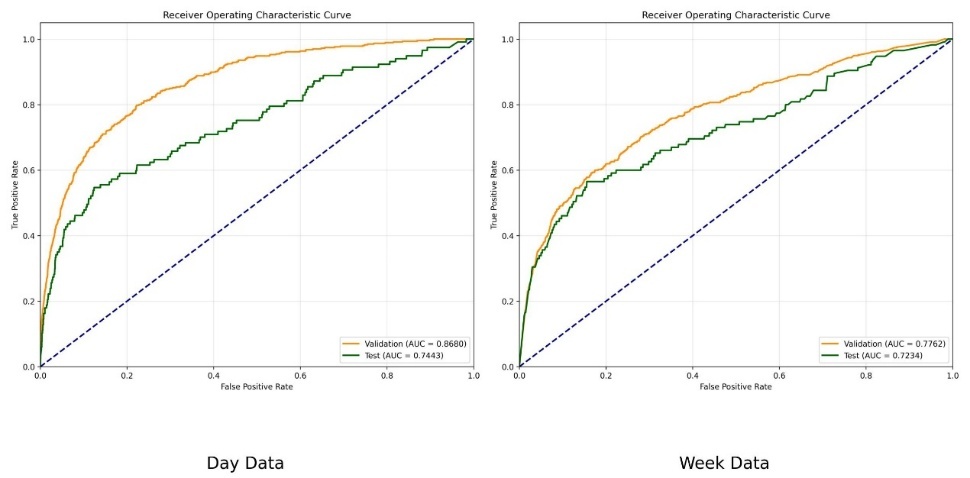


Figure 7. ROC-AUC curve of day (left) and week (right) data.

* The training-test AUC difference for daily data is 14.25%, and for weekly data is 5.38%, reflecting more severe overfitting in the daily data model, which correlates with its high feature dimensionality of 70 dimensions and fine time scale.
* An ideal model should approach the upper left corner (TPR=1, FPR=0), while the current model distributes near the diagonal, indicating room for improvement. The daily data model's AUC (0.8680) exceeds that of weekly data (0.7762), showing that fine-grained temporal features are more effective for injury prediction.

## 8.4 Average AUC Training Curve

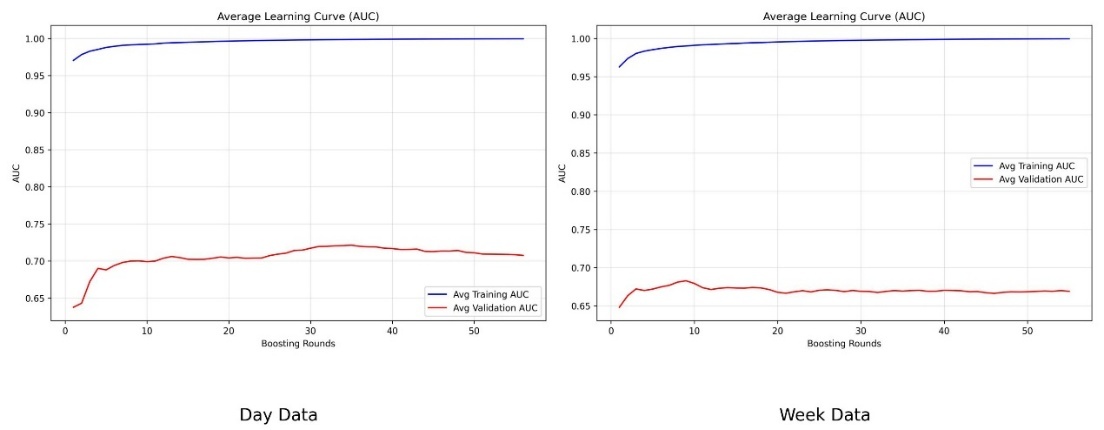


Figure 8. Average AUC training curves of day (left) and week (right) data.

For day data, the training AUC of daily granularity data with 70-dimensional temporal features sharply jumps from 0.97 to 1.0 in 50 rounds, which confirms the strong memory ability of LightGBM for fine-grained features. However, the validation AUC only climbed from 0.65 to 0.7 and then stagnated, forming a gap of 0.3 with the training AUC, exposing key contradiction. The synthetic samples generated by SMOTE make the model overfit the local time series noise, and lose the generalization power.Daily dynamic metrics (e.g., perceived\_recovery) are overweighted, resulting in discriminant collapse when new data shifts (test set recall drops 21.7%).

For week data, the training AUC on weekly granularity data with 69-dimensional statistical features consistently skyrocketed to 1.0, demonstrating a strong fit to the aggregate metric (e.g., nr.sessions.2\_std\_7d). However, the validation AUC is only 0.68, and the whole fluctuation is ± 0.02, revealing deeper problems. Weekly averages (e.g., total km\_max\_28d) smooth out key discriminative details, forcing the model to rely on spurious statistically relevant features. Feature dimension compression (69 vs 70 dimensions) instead amplifies random fluctuations, resulting in lower and unstable validation AUCs.

## 8.5 Binary-logloss Training Curve

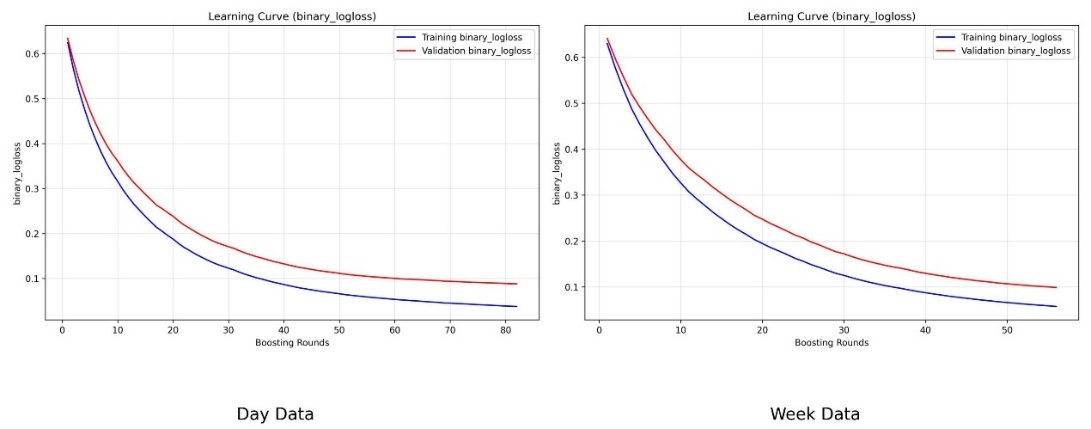


Figure 9. Binary-logloss training curves of day (left) and week (right) data.

For day data, the training binary\_logloss for the daily granularity data (70-dimensional features) starts at 0.6 and rapidly decreases to 0.05 with the number of lifting rounds (stable after about 100 rounds), reflecting the strong fitting ability of LightGBM for fine-grained time series (training AUC≈1.0). However, the validation binary\_logloss only slowly decreases from 0.6 to 0.1 and then stagnates, resulting ina 10x gap with the training loss. This is consistent with the test set recall plummeting by 21.7% (0.7532→0.5897), which reveals core problems. Synthetic samples generated by SMOTE alleviate class imbalance, but allow the model to learn local patterns that are not generalizable (e.g., noise patterns for specific athletes). We rely too much on daily training details, causing prediction confidence to collapse when the temporal pattern shifts in new data

For week data, the training binary\_logloss on weekly granularity data (69-dimensional aggregate features) is synchronously reduced to 0.05, proving that the model's ability to fit statistical features such as nr.sessions.2\_std\_7d is not impaired. However, the validation binary\_logloss stagnates at ≥0.1 and continues to fluctuate slightly, reflecting that weekly averages (e.g., total km\_max\_28d) smooth out critical details (e.g., one-day load spikes), forcing the model to rely on spurious correlations, and feature dimension compression (69 vs 70 dimensions) fails to suppress overfitting; conversely, the validation loss is higher and less stable due to the lack of discriminative details (compared with day data).

# 9. Reference

[1] Ke G, Meng Q, Finley T, et al. Lightgbm: A highly efficient gradient boosting decision tree[J]. Advances in neural information processing systems, 2017, 30.

[2] Bai S, Kolter J Z, Koltun V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling[J]. arXiv preprint arXiv:1803.01271, 2018.

[3] Chawla N V, Bowyer K W, Hall L O, et al. SMOTE: synthetic minority over-sampling technique[J]. Journal of artificial intelligence research, 2002, 16: 321-357.

[4] Muntasir Nishat M, Faisal F, Jahan Ratul I, et al. A Comprehensive Investigation of the Performances of Different Machine Learning Classifiers With SMOTE‐ENN Oversampling Technique and Hyperparameter Optimization for Imbalanced Heart Failure Dataset[J]. Scientific Programming, 2022, 2022(1): 3649406.

[5] Dalianis H, Dalianis H. Evaluation metrics and evaluation[J]. Clinical Text Mining: secondary use of electronic patient records, 2018: 45-53.

[6] Yacouby R, Axman D. Probabilistic extension of precision, recall, and f1 score for more thorough evaluation of classification models[C]//Proceedings of the first workshop on evaluation and comparison of NLP systems. 2020: 79-91.

[7] Lövdal, S., den Hartigh, R., & Azzopardi, G. (2021). Injury Prediction in Competitive Runners with Machine Learning. International journal of sports physiology and performance, 16(10), 1522–1531. <https://doi.org/10.1123/ijspp.2020-0518>

[8] Gabbett TJ. Br J Sports Med Published Online First: [please include Day Month Year] doi:10.1136/ bjsports-2015-095788

[9] Journal of Sport & Exercise Psychology, 2009, 31, 60-76 2009 Human Kinetics, Inc.