# EI327-1-工程实践与科技创新 IV-I

运动探测 5-3 组

陈浩南，余北辰，管仁阳

**摘要**

通过收集和分析各类运动传感器（陀螺仪、加速度计）上的数据，我们可以得到目标对象的运动信息，这类运动检测任务具有广泛的研究价值和应用前景。在本次工程实践与科技创新项目中，本小组基于传统的运动检测算法，对其进行了改进创新和进一步的研究。本项目基于 Motion Sense 数据集展开，主要由**模型优化**与**数据特性探究**两方面的工作。在模型方面，本小组复现了 CNN+LSTM 模型，并尝试了多种集成学习模型，如 GBDT，LightGBM，RondomForest，KNN 以及其他机器学习模型以提升模型的性能。在数据方面，通过更改时间片长度，修改实验数据有效率以及传感器屏蔽等方法，获得了数据量有效性对学习结果的影响程度以及不同类型传感器间的数据共享程度相关的结论。

**关键词**：运动检测、机器学习

## 1 概述

随着越来越多的嵌入式运动传感器被应用于日常生活中，如何利用这些信息来完成对运动的检测分类是当下学术界所关心的热门问题之一。运动传感器，如加速度计和陀螺仪，以三维的方式测量设备的瞬时加速度和旋转。来自嵌入在便携式和可穿戴设备中的运动传感器的原始数据流（1D 时间序列数据）可以用来进行活动识别。在本次工程实践的过程中，本小组的主要任务为：通过运动传感器的多个原始数据流对运动类型进行分类。

首先，本小组进行了相应的文档调研，查阅了基于运动传感器数据的运动分类、通过时频域转换或直接使用时域数据等不同处理方式各自的优劣等方面的内容，最终选择了 [2] 和 [3] 这两篇论文作为本项目的主要参考论文。之后，基于开源的运动传感器数据集 MotionSense, 对原始数据进行降噪、滤波等预处理操作，使后续的机器学习可以获得更好的实验效果。在机器学习方面，我们首先复现了参考论文 [2] 中用于运动检测的神经网络模型，并测试了实验效果。在此基础上，我们对神经网络的结构进行了改进和创新，尝试了多种当下热门的机器学习模型，如 GBDT，LightGBM，RondomForest，KNN 以及集成学习模型等。通过实现并尝试新的分类模型，我们评估了不同方法的性能差别，并在原始模型的基础了提高了运动检测的准确率和分类效率。最后，在指导老师的帮助下，我们对数据集的内在规律进行研究，通过更改时间片长度，修改实验数据有效率以及传感器屏蔽等方法处理 MotionSense 数据集，并将其输入不同的机器学习模型中，观察实验效果。最终得出结论，获得了数据量有效性对学习结果的影响程度以及不同类型传感器间的数据共享程度。

本次项目的工作可概况为以下三个阶段：

1. 对参考论文 [2] 和 [3] 的 CNN+LSTM 模型进行了**复现**，并**比较 CNN 和 LSTM 模型的时序表达能力**。
2. 实现并尝试了**多种不同的机器学习模型**在运动检测方面的效果，达到了**远超 CNN+LSTM 方法的准确率 (97.8%)**。

3. 探究 Motion Sense 运动探测数据集的**数据内在特性（1. 单个输入包含的时间点数量、2. 数据有效率、3. 数据集中性、4. 传感器重要程度）**。

输入特征的“**有效率**”指的是**特征向量中的每个数值不被置为 *0* 的概率**。例如，对于输入特征 $[a, b, c, d]$ 在有效率为 0.25 时，$a, b, c, d$ 各自以 0.25 概率置为 0。假设 $a, b, c$ 被置为 0，那么模型的输入就是 $[0, 0, 0, d]$，无效数据并不会被压缩。

在第一阶段，本小组比较了 CNN+LSTM 模型、CNN 模型与 LSTM 模型在训练数据效果与它们的泛化性能，比较了 CNN 与 LSTM 模型对时序数据的表达能力差异。

第二阶段主要是工程实现。本小组使用了梯度提升决策树（GBDT）、随机森林、K 近邻聚类和 Ensemble Selection 四类、13 种机器学习模型来进行运动分类，最高准确率达到 **97.8**%。

在第三阶段，本小组设计实验探究 1. 一次输入包含的时间点数量、2. 数据有效性、3. 数据集中性，对训练效果的影响，与传感器的重要程度。

# 2 相关工作

本次工程实践与科技创新的内容主要参考了两篇文章:其一是伦敦玛丽女王大学的 Malekzadeh 等人在 2019 年发表的文章 Mobile sensor data anonymization[1]，其二为诺丁汉特伦特大学的 Eiman Kanjo 等人在 2019 年发表的文章. Deep learning analysis of mobile physiological, environmental and location sensor data for emotion detection[2]。在这一小节中，我们将对这两篇相关工作的具体方法、内容及其贡献进行说明与介绍。

Mobile sensor data anonymization[1] 这篇文章的主要关注点在于移动传感器所产生的数据可能带来的用户隐私信息的泄露问题，智能运动手环上的运动传感器产生的用户运动数据有可能泄露用户的身高、体重、性别等敏感的隐私信息。文章的作者针对该问题提出了对于移动传感器的数据匿名化方法。具体而言，作者在信息论的范畴内定义了运动传感器的匿名性问题，并设计了一套编码与解码的方案，通过多目标损失函数训练深度自动编码器，以达到最大化活动识别准确率的同时最小化用户识别准确率的多目标效果，实现实用性与隐私性上的相对平衡。

Deep learning analysis of mobile physiological, environmental and location sensor data for emotion detection[2] 这篇文章的关注点是人类情感的监测与分类。与一些通过嵌入特征工程过程来保障情感建模的传统的机器学习方法相比，该工通过添加和删除来自不同模式的大量传感器信号，通过一个迭代过程进行情感分类。该工作使用混合模型，将三种传感器模式的局部互动合并到全局模型；该工作同时通过一系列的学习方法，消除了对人工特征提取和工程的需求。经过实验验证，使用该工作提出的 CNN-LSTM 混合模型，相较于传统的嵌入特征工程的方法以及经典的全连接深度神经网络准确率都更有保障。

我们的工作是基于这两篇文章而展开的。首先，我们的工作采用 Mobile sensor data anonymization[1] 这篇文章的数据集作为模型的输入，预期实现在该数据集上的运动数据识别的功能；另一方面，我们将 Deep learning analysis of mobile physiological, environmental and location sensor data for emotion detection[2] 这篇文章中所提出的混合模型进行复现，并且将其当作本次实践的 baseline 模型，探究其在目标数据集上的表现，并且将其与一些常用的集成学习模型进行效果上的对比。

# 3 模型

## 3.1 Baseline

我们以 Deep learning analysis of mobile physiological, environmental and location sensor data for emotion detection[2] 这篇文章中的 CNN-LSTM 模型作为本次实践的 Baseline。在本节中，我

们将对此模型进行详细的介绍。

CNN 是深度学习领域内应用最为广泛的模型之一，能够快速地提取输入的运动数据的空间特征。但是由于 CNN 对输入数据的顺序缺乏敏感性，无法提取数据的时间特征，Baseline 模型在 CNN 层后连接 LSTM 神经网络，以进行对输入的运动数据的时间信息进行提取。将 CNN 和 LSTM 结合，保证了网络可以从输入的运动数据序列中学习到空间特征和时间相关性。

### 3.1.1 CNN

CNN（卷积神经网络）是一种深层前馈网络，是一种有效的复杂非线性模型建模工具，在对多阵列数据（例如时间序列、图像等）进行处理时效果突出。CNN 由卷积层、池化层和完全连接层构成，不同层功能各异。

CNN 模型的具体结构如图1所示：

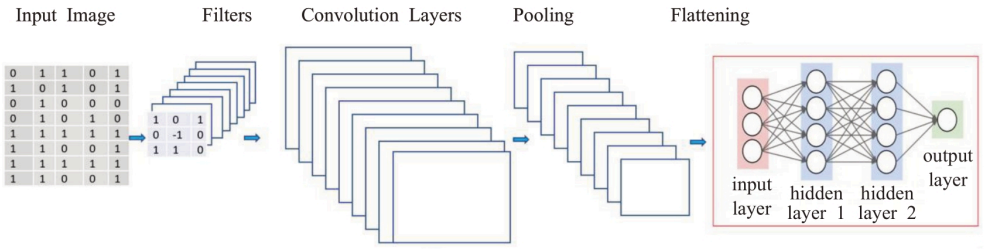



图 1: CNN architecture

卷积层通过卷积核在输入向量上卷积，生成特征向量；池化层分为最大池化和均值池化，池化层可以减少特征向量和 CNN 参数的大小，减少训练时间和内存需求，并控制过度拟合；完全连接层用于将输入转换为向量，并实现不同任务。

对于 CNN 而言，以下几点是其最主要的优势：

1. 可以共享卷积核（共享参数），使得 CNN 对高维的数据处理没有压力;
2. 不需要选择特征属性，只需要训练好每一个权重，也就可以得到特征值;
3. 在深层次的神经网络中，对信息的抽取更为丰富，从而使得最终表达效果更好。

### 3.1.2 LSTM

LSTM（长短时记忆网络）是 RNN（循环神经网络）的一种变体。为了解决梯度消失的问题，并且增强模型的泛化性，LSTM 在 RNN 的基础上进行了一系列的优化。与 RNN 相比，LSTM 主要将门与状态单元的概念引入其中，在数据分析中相较于 RNN 有更好的适应性。

下面的图2是 LSTM 的网络单元结构示意图。

LSTM 网络由遗忘门、输入门和输出门这三个基本结构组成。遗忘门的作用是定义要遗忘的信息，即定义从上一个细胞状态中所需要删除的信息。数据经遗忘门筛选历史信息后，经过输入门，由输入门确定所需要更新的新生信息。经过遗忘门和输入门实现对细胞状态的更新后，由输出门选择当前状态所需输出的信息。

LSTM 通过隐藏层间的输入输出在处理关联数据时，能有效存储数据特性并解决梯度消失的问题。通常用于研究输入信息在时间维度上的相关性问题，在处理时间序列数据有着很大优势。

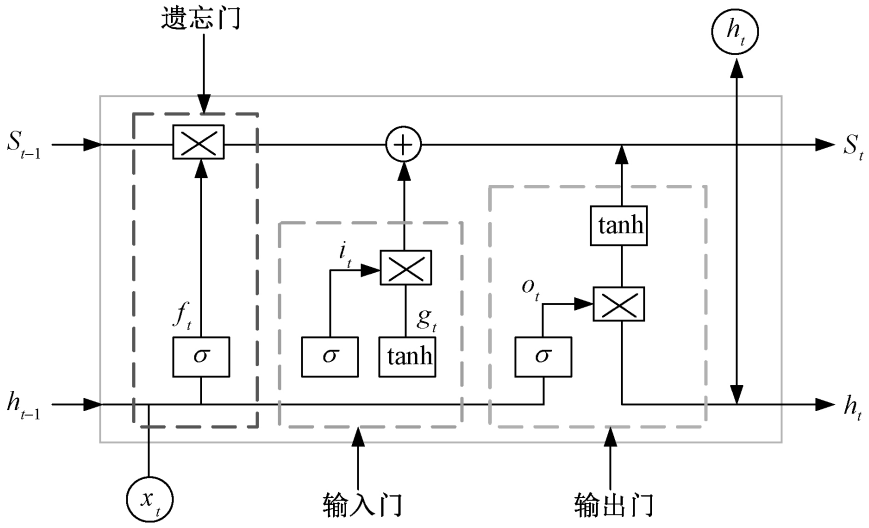



图 2: LSTM architecture

### 3.1.3 CNN-LSTM

Baseline 使用的模型结构是 CNN-LSTM 的结构。其结构在 CNN 的基础上，增加了一个 LSTM 层，如图3所示。数据的时间维度在卷积操作中进行预处理，其产生的完全连接层被送入 LSTM 单元中。随着时间的推移，LSTM 单元会根据其当前的输入和其内部状态的历史记录，来输出、覆盖或重置其内部存储器。

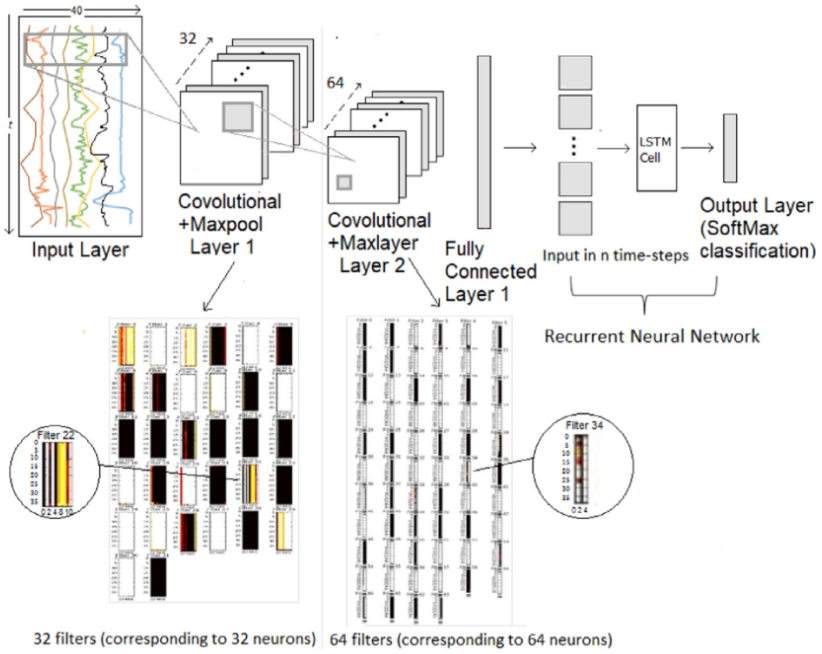



图 3: CNN-LSTM architecture

CNN-LSTM 模型在 CNN 的基础上，通过 LSTM 对的运动数据的时序信息进行提取和处理，弥补了 CNN 对输入信息的先后顺序缺乏敏感性的不足，保障了模型的效率以及可靠性。

## 3.2 集成学习模型

### 3.2.1 Gradient Boosting Decision Tree

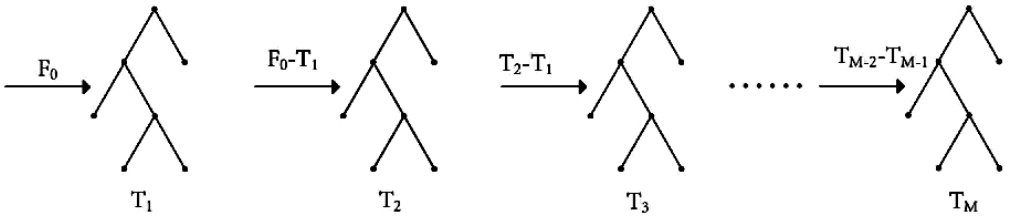



图 4: Gradient Boosting Decision Tree model

**模型简介:** GBDT(Gradient Boosting Decision Tree) 的中文名称为 “梯度提升决策树”，在 1999 年由 Jerome Friedman[3] 提出，是一种迭代的决策树算法。该算法由多棵决策树组成，且每一棵决策树的规模都不大。进行模型预测时，对于输入的样本，首先赋予其一个初值，之后对每一棵决策树进行遍历，在此过程中，每棵决策树对预测的结果进行调整与修正。最后，将每棵决策树的结果进行累加求和，输出最终的预测结果。GBDT 算法在处理多特征输入分类与回归问题上表现优异，模型训练速度快，精度高。正因为如此，我们小组尝试在运动检测的问题上使用 GBDT 算法，并对实验结果进行记录、观察和分析。

**模型结构:** GBDT 算法其核心思想在于梯度优化。在训练时，每一棵决策树是从之前所有树的残差中进行学习的。每一颗决策树 $T_m$ 在生长时，选择分枝增益最大的方向，而回归树 $T_m$ 的训练目标则是真实值与 $T_1 + T_2 + T_3 + \cdots + T_{m-1}$ 的残差。

下面对 GBDT 算法的主要执行过程进行分步介绍:

- 设置初始的 $F_0(x) = \arg\min \sum_{i=1}^{N} L(y_i, \gamma)$ , 其中 $L$ 为树的代价函数。
- 对于第 $m$ 棵树，计算其负梯度，其中 $i = 1, 2, 3, \cdots, N$，$N$ 为样本数。

$$\widetilde{y_i} = -\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}$$

- 第 $m$ 棵树的参数:

$$\omega_m = \arg\min_{\omega} \sum_{i=1}^{N} (\widetilde{y_i} - h(x_i : \omega))^2$$

  其中 $h$ 是树函数。

- 最优化第 $m$ 棵树的权重:

$$\gamma_m = \arg\min_{\gamma} \sum_{i=1}^{N} L(y_i, F_{m-1}(x_i) + \gamma h(x_i : \omega_m))$$

- 最后，将当前树与已训练完成的树合并:

$$F_m(x) = F_{m-1}(x) + \gamma_m h(x : \omega_m)$$

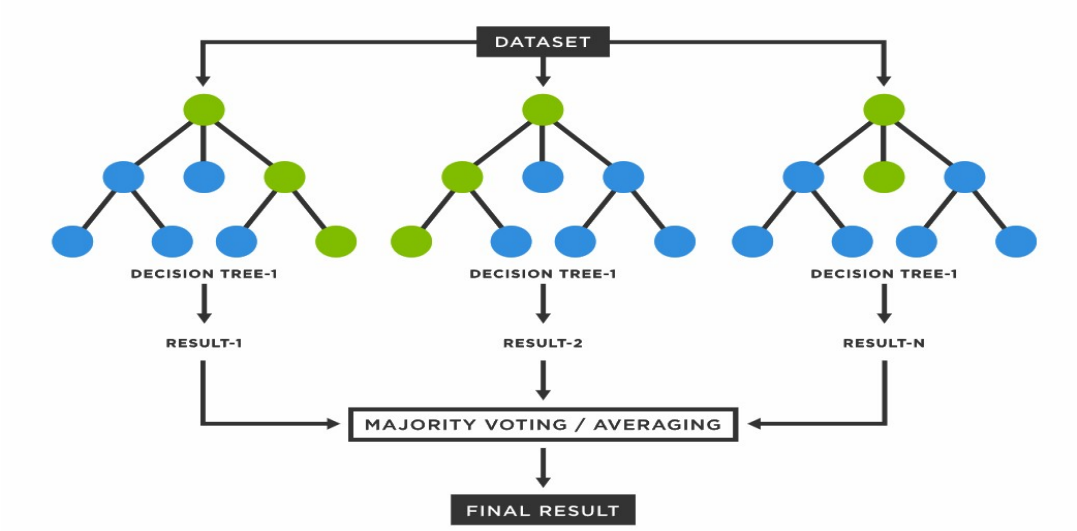



图 5: Ensemble model

### 3.2.2 Extremely Randomized Trees and Random Forests

**模型简介:** 随机森林 (Random Forests) 是一种有监督学习算法，是以决策树为基础学习器的集成学习算法。随机森林代码简单，易于实现，计算开销也很小，但是它在分类和回归上表现出非常惊人的性能。因此，在本次实验中，我们小组尝试将随机森林模型应用到运动检测这一课题上，并观察实验结果。

随机森林是 Bagging 的一个扩展变体，在以决策树作为基学习器构建 Bagging 集成的基础上，进一步在决策树的训练过程中加入了随机属性的选择。具体来说，传统决策树在选择划分属性时是在当前结点的所有候选属性（假定有 d 个）中选择一个最优属性；而在 RF 中，对基决策树的每个结点，先从该结点的候选属性集合中随机选择一个包含 k 个属性的子集，然后再从这个子集中选择一个最优属性用于划分。由此，随机森林的基学习器的“多样性”不仅来自样本的扰动，还来自属性的扰动，使得最终集成的泛化能力进一步增强。

**模型结构:** 随机森林之所以被称为“森林”，是因为它生成了决策树森林。然后，来自这些树的数据合并在一起，以确保最准确的预测。虽然单独的决策树只有一个结果和范围狭窄的群组，但森林可以确保有更多的小组和决策，从而获得更准确的结果。它还有一个好处，那就是通过在随机特征子集中找到最佳特征来为模型添加随机性。本次实验所使用的随机森林模型如图5所示。具体的代码使用了 sklearn.ensemble.RandomForestClassifier 模块进行实现，下面将介绍模型的结构和超参数选择。

- criterion 不纯度的衡量指标，有基尼系数和信息熵两种选择
- max_depth 树的最大深度，超过最大深度的树枝都会被剪掉
- min_samples_leaf 一个节点在分枝后的每个子节点都必须包含至少 min_samples_leaf 个训练样本，否则分枝就不会发生
- min_samples_split 一个节点必须要包含至少 min_samples_split 个训练样本，这个节点才允许被分枝，否则分枝就不会发生。
- n_estimators 森林中树木的数量，即基评估器的数量。这个参数对随机森林模型的精确性影响是单调的，n_estimators 越大，模型的效果往往越好。但是相应的，任何模型都有决策边界，n_estimators 达到一定的程度之后，随机森林的精确性往往不在上升或开始波动，并且，

n_estimators 越大，需要的计算量和内存也越大，训练的时间也会越来越长。对于这个参数，我们是追求的是在训练难度和模型效果之间取得平衡。

### 3.2.3 K Neighbour Cluster

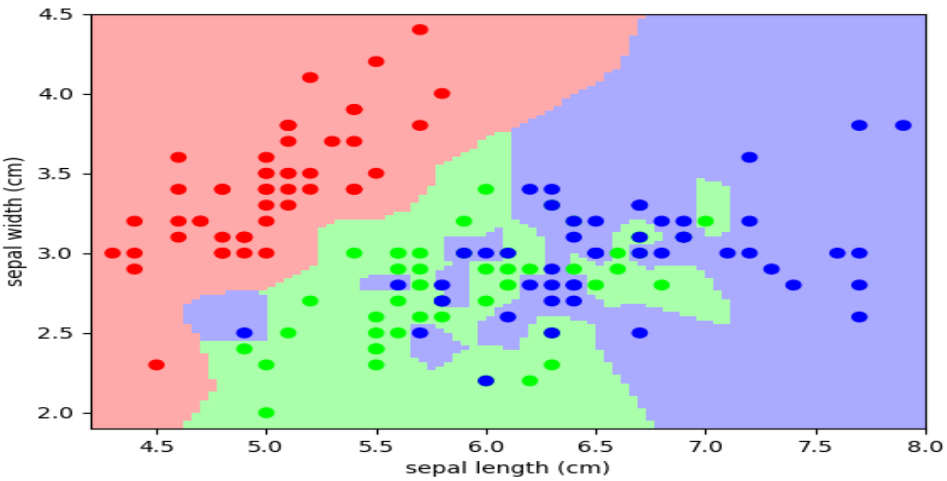



图 6: KNN model

**模型简介：**观察到在本次运动检测的实验中，存在一个测试对象同时与多个训练对象匹配，导致一个训练对象被分到了多个类的问题，基于这种问题，本小组决定采用 KNN 的分类方法来尝试降低误差，提高准确率。与传统的分类器将全部的训练数据所对应的类别都记录下来，当测试对象的属性和某个训练对象的属性完全匹配时，便可以对其进行分类不同，kNN 是通过测量不同特征值之间的距离进行分类。

K Neighbour Cluster 的大致思路是：如果一个样本在特征空间中的 k 个最相似 (即特征空间中最邻近) 的样本中的大多数属于某一个类别，则该样本也属于这个类别。KNN 算法中，所选择的邻居都是已经正确分类的对象。该方法在定类决策上只依据最邻近的一个或者几个样本的类别来决定待分样本所属的类别。

**模型结构：**在确定了算法框架之后，本小组实现了 KNN 模型并将其用于本次运动检测的任务。下面对于模型结构和模型的超参数选择进行简要介绍。

- 超参数 K 的选择：k 近邻算法中的 k 值选取对于模型的准确率影响至关重要。当 k 值过小时，会引发过拟合的问题，导致模型经过训练后易学习到噪声信息而忽略的数据集中原有的规律。从而导致模型的准确度降低。而当 k 值选择过大时，就相当于用较大邻域中的训练数据进行预测，这时与输入实例较远的（不相似）训练实例也会对预测起作用，使预测发生错误，k 值的增大意味着整体模型变得简单，同样也会导致模型的性能下降。在本次实验中，本小组采用了交叉验证的方法不断调整超参数的取值，并选择性能最高的 K 值作为最终的取值。

- 数据点间距离的计算：在 KNN 算法中，通过度量数据点和其他样本之间的距离来实现分类的效果。而如何度量数据点之间的距离则是重中之重。当下，采用的度量方法主要有欧式距离、曼哈顿距离两种。具体的计算公式如下：

$$\text{欧式距离:} L_2(X_i, X_j) = \sqrt{(\sum_{l=1}^{n}(\left|X_i^{(l)}, X_j^{(l)}\right|)^2}$$

$$\text{曼哈顿距离:} L_1(X_i, X_j) = (\sum_{l=1}^{n}(\left|X_i^{(l)}, X_j^{(l)}\right|)$$

在本次实验中，本小组使用了更为常见的欧氏距离来度量样本点之间的距离。

- 特征归一化处理：观察数据后本小组发现，因为由于各个传感器间量纲不同，导致数值的范围差距较大。因此可能会导致在度量距离时不同维度上数据信息的贡献度不同，从而忽略了数据的内在规律。所以本小组在进行模型的训练和测试之前，首先对不同纬度上的传感器数据进行了归一化处理。首先选择每一轴上的最大值和最小值之差:$M_j = \max_{i=1,\ldots,m} X_{ij} - \min_{i=1,\ldots,m} X_{ij}$，之后在计算距离时将每一个纬度的数值除以 $M_j$ 以进行归一化，具体公式如下：

$$d_2(X_i, X_j) = \sqrt{(\sum_{l=1}^{n}(\left|\frac{X_i^{(l)}}{M_j}, \frac{X_j^{(l)}}{M_j}\right|)^2}$$

### 3.2.4 Ensemble Selection from Libraries of Models

**模型简介：** 在使用有监督的机器学习方法来解决运动检测这一实际问题的过程中，本小组尝试了多种不同的机器学习模型（具体内容请见上文），并期望在其中选择一个具有较好且相对稳定的预测效果的模型。除了选择单一的机器学习模型之外，本小组尝试采用集成学习的方法，组合多种实验模型进行动作的分类。

Ensemble 的方法就是组合多个模型以期望得到一个较优的结果。换句话说，Ensemble 的方法就是组合许多弱模型 (weak learners，预测效果一般的模型) 以得到一个强模型 (strong learner，预测效果好的模型)。Ensemble 中组合的模型可以是同一类的模型，也可以是不同类型的模型。

Ensemble 方法对于大量数据和不充分数据都有很好的效果。因为一些简单模型数据量太大而很难训练，或者只能学习到一部分，而 Ensemble 方法可以有策略的将数据集划分成一些小数据集，分别进行训练，之后根据一些策略进行组合。相反，如果数据量很少，可以使用 bootstrap 进行抽样，得到多个数据集，分别进行训练后再组合。

使用 Ensemble 的方法在评估测试的时候，相比于单一模型，需要更多的计算。因此，有时候也认为 Ensemble 是用更多的计算来弥补弱模型。

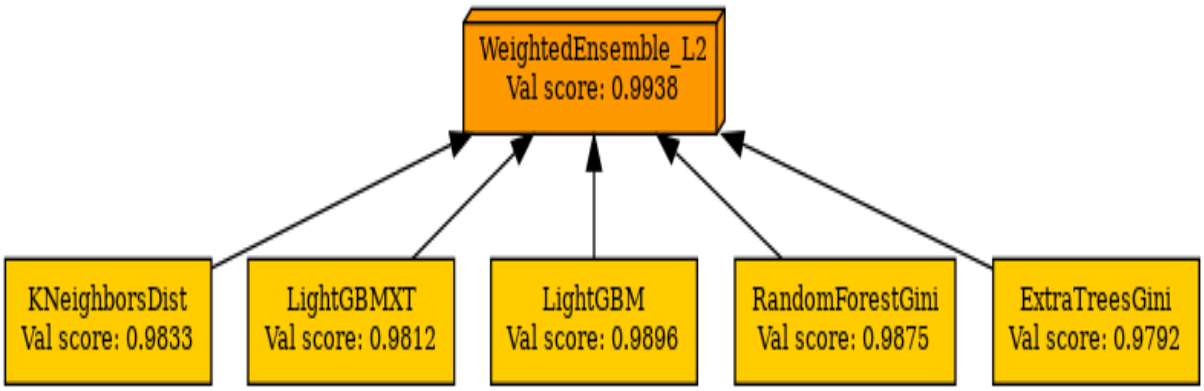



图 7: Ensemble model

**模型结构：** 当下对不同模型的集成方法主要有：bagging、boosting、stacking 这三种方式。其中利用 bagging 方法集成决策树的方法就是我们在上文中讨论过的随机森林方法。而基于 boosting 的提升方法我们也在上文中通过 LightGBM XGboost, CatBoost 等模型进行了尝试。因此，本小组最终决定依据参考论文 [4] 所提出的集成学习算法，选择上文中实现的部分模型进行集成以实现最终的分类任务。

1. Start with the empty ensemble.
2. Add to the ensemble the model in the library that maximizes the ensemble’ s performance to the error metric on a hillclimb (validation) set.
3. Repeat Step 2 for a fixed number of iterations or until all the models have been used.
4. Return the ensemble from the nested set of ensembles that has maximum performance on the hillclimb (validation) set.

通过上述的集成学习方法，本小组构建出了集成学习的最终结构图，如图 7所示。

# 4 数据集

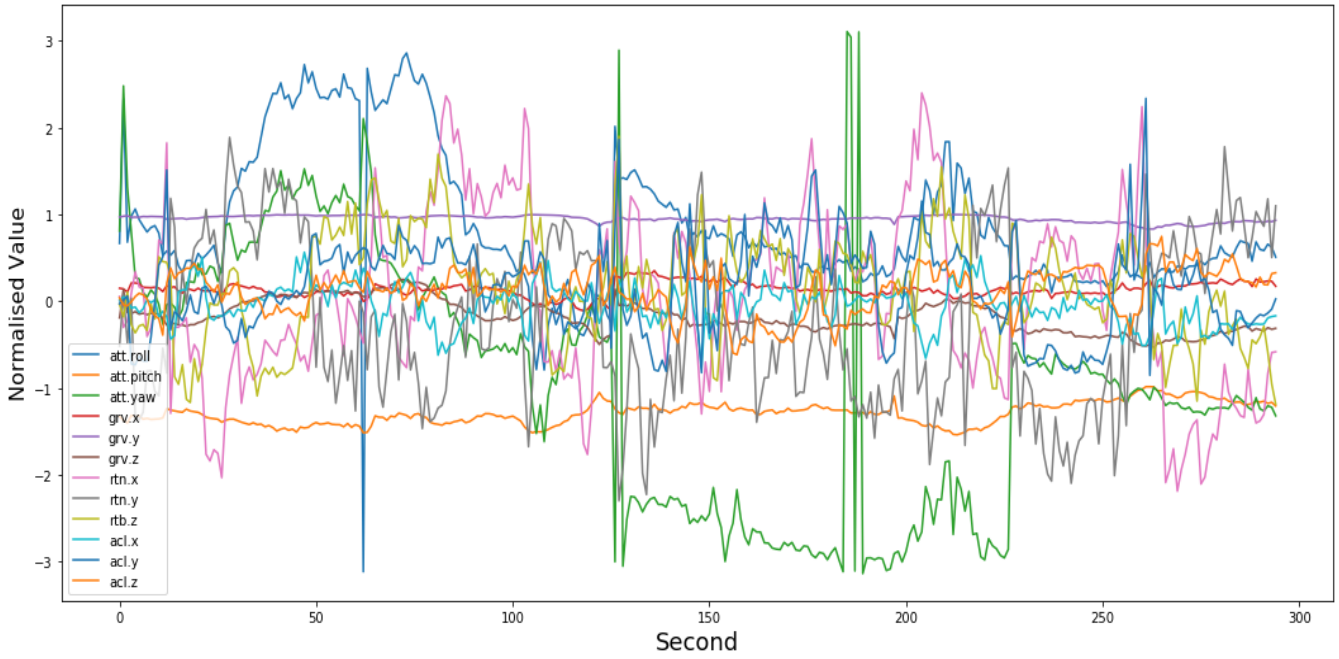
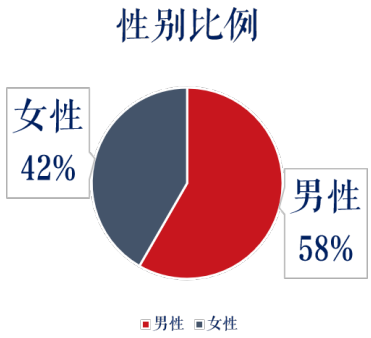



图 8: Motion Sense 数据集

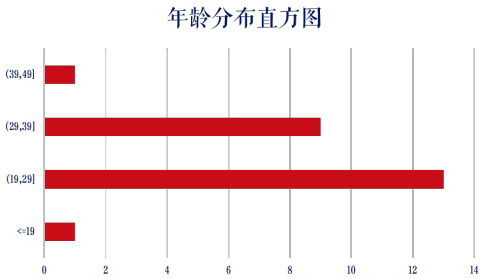
本次实践采用的数据集为公开数据集 Motion Sense。该数据集的内容包括由加速度计、陀螺仪这两个传感器所产生的时间序列数据，包括姿态、重力、加速度以及旋转速度。该数据集中的所有数据都是通过 iPhone6s 所装载的 Core Motion 框架在 50Hz 的采样率下收集。此数据集的采集由 24 位不同年龄、性别、体型的志愿者在同一环境分别进行，每一位志愿者分别进行 6 种不同的运动：下楼、上楼、步行、慢跑、坐下、站立，每一种运动重复实验 15 次。

图8是该数据集的一个示例的可视化，志愿者进行步行活动时所对应的传感器产生的 12 种特征所对应的时间序列。

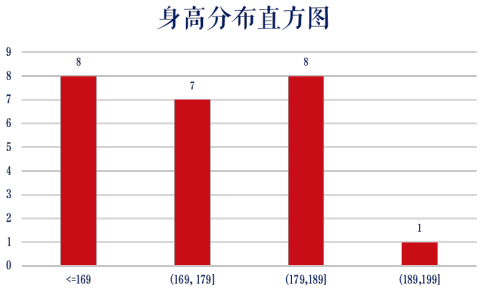
Motion Sense 数据集中志愿者的年性别、年龄、身高、体重分布如图 9。其中男性的比例略大于女性；年龄主要分布在 19-39 岁这一区间，几乎没有 40 岁以上人群的样本数据，本小组认为这会导致数据的同质化比较严重。样本的体重近似正态分布，身高分布也较为均匀。



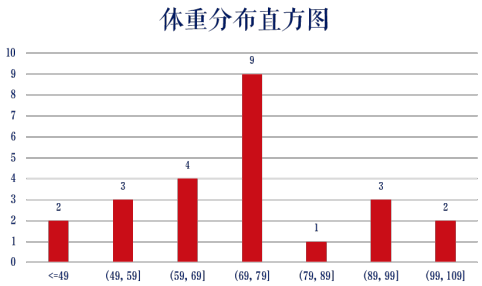

(a) 性别比例




(b) 年龄分布




(c) 身高分布




(d) 体重分布

图 9: 性别、年龄、身高、体重分布

# 5 实验分析

## 5.1 实验设定

本项目基于 Motion Sense 数据集 [1]，设计了一系列实验用以完成如下任务：

1. 将 **Baseline 模型（CNN+LSTM）、CNN、LSTM** 各自效果进行比较。
2. 探究**单个输入的数据包含的时间点数量**对整体效果及推理时间的影响。
3. 使用多种机器学习算法实现**最优效果**。
4. 探究**数据有效率**对整体效果的影响。
5. 探究四种**传感器**（加速度、旋转速度、重力方向、位姿方向）重要程度。

本项目中的 Baseline 模型，也即 CNN 和 LSTM，都使用 PyTorch 实现，采用 Adam 优化器，损失函数为对数似然代价函数，学习率为 0,01 并使用 PyTorch 框架中的 Scheduler 自动优化学习率。

本项目中的集成学习方法使用机器学习框架 MxNet 中的内置库，MxNet 框架会使用网络架构搜索（NAS）的方法进行超参数配置、基模型选择与模型优化。本项目探究的集成学习方法主要有梯度提升决策树（GBDT）、随机森林（Random Forest）和极端随机树（Extremely Randomized Trees）、K 近邻聚类（K Neighbor Clustering）、Ensemble Selection 四种方法，每种方法都有多个具体实现，如图 10所示。

其中，GBDT 的方法采用了业界常用的 XGBoost、LightGBM 与 CatBoost 三种开源实现，K 近邻聚类采用了使用基尼系数与熵的两个版本。

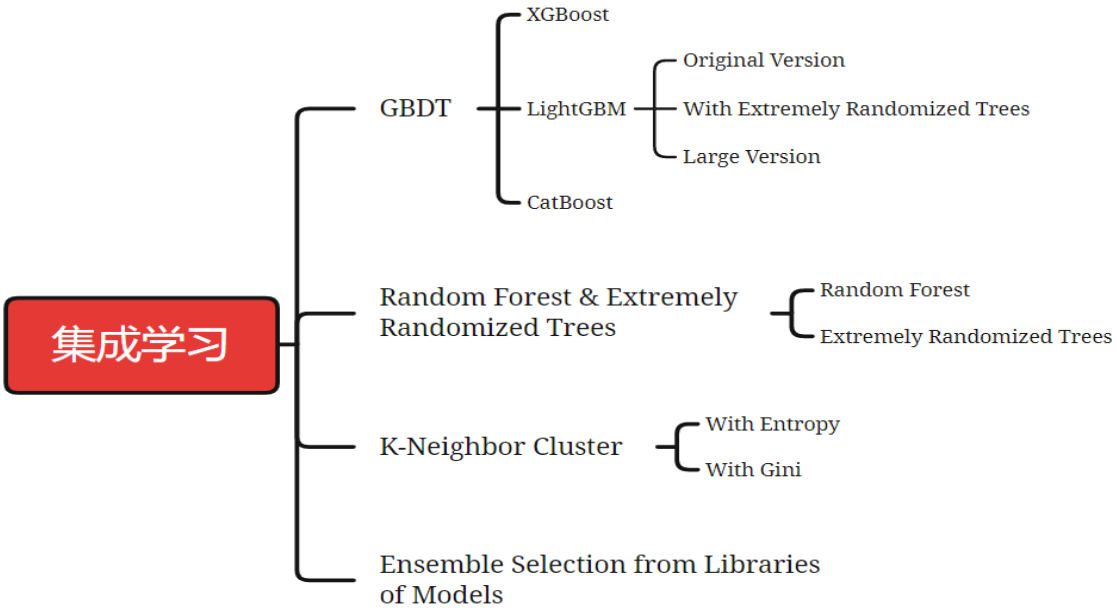



图 10: Ensemble model

## 5.2 Baseline

本项目设计了三组实验来探究 Baseline 模型，分别为 CNN+LSTM 模型、CNN（无 LSTM）模型、LSTM（无 CNN）模型。其中 CNN 模型具体结构如图 11所示。

```
MotionSenseModel(
  (conv1): Conv2d(1, 32, kernel_size=(3, 3), stride=(1, 1), padding=same)
  (conv2): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1), padding=same)
  (conv2_drop): Dropout2d(p=0.5, inplace=False)
  (fc1): Linear(in_features=576, out_features=120, bias=True)
  (lstm): LSTM(10, 10, batch_first=True)
  (fc2): Linear(in_features=10, out_features=6, bias=True)
)
```

图 11: Baseline 模型结构

三组实验中统一取一次输入的时间片（Time_Slice）为 12，效果汇总于表 1。

由于 Motion Sense 数据集的数据内在结构较为简单，在此数据集很容易体现出模型参数对过拟合、欠拟合的影响。此外，该实验还验证了 LSTM 对于时序数据的更好的表征能力。

**拟合性** 从此组实验中可以看到模型参数量最大的完整模型 CNN+LSTM 在训练集上的效果最好；而当仅使用 LSTM 时模型取得了更好的训练数据准确度，但是泛化能力有所下降；仅使用 CNN 模型在训练集上表现出了严重的过拟合，相比于 CNN+LSTM，CNN 在训练数据的准确率方面提高了 3%，而测试数据准确率（泛化能力）下降了 3.6%。

**LSTM 的时序表达能力** 本实验也验证了 **LSTM 模型对时序数据的表达能力强于 CNN**。LTM 模型本身的参数量并不大，仅有 CNN+LSTM 模型的 20%、CNN 模型的 21%，但是它的效果却

| 模型 | 模型参数量 | 测试集准确率 | 验证集准确率 | 测试集 Loss | 训练集 Loss |
|---|---|---|---|---|---|
| CNN+LSTM | 89002 | 93.3% | 96.3% | 0.32 | 0.09 |
| CNN | 88782 | 89.8% | 98.9% | 2.99 | 0.01 |
| LSTM | 18346 | 92.3% | 97.2% | 0.35 | 0.08 |

表 1: Baseline 模型结果

与 CNN+LSTM 相近。**去掉 LSTM 以后，虽然 CNN 与 CNN+LSTM 模型的参数量相当，但是 CNN 模型表现出了较强的过拟合**。

## 5.3 集成学习模型

### 5.3.1 时间点

本实验选取一次输入的时间点有 12，24，36，48，60，72（受制于训练设备性能，无法选取时间点大于等于 84）。不同的输入时间点数量下的最优模型与测试集准确率汇总与表 2。

| Time Slice | 最优模型 | 测试集准确率 |
|---|---|---|
| 12 | XGBoost | 94.7% |
| 24 | Weighted Ensemble | 96.5% |
| 48 | LightGBM（Extremely Randomized Trees） | 97.8% |
| 72 | LigthGBM (Large) | 97.2% |

表 2: 不同的输入时间点数量对效果的影响

**集成学习模型下输入特征数量对拟合效果的影响** 通常认为一次输入更多时间点的数据意味着更多输入特征，也就意味着更丰富的输入信息，会带来更好的预测效果。然而，更多的输入特征也必须要使用更大的、表达能力更强的模型来提取。从表 2中的数据可见，随着 Time Slice 增大，为达到最优效果所需要的模型规模也不断增大。而本实验中使用的集成学习模型的模型规模是有上界的，当 $Time\_Slice > 48$ 时，最大的模型也无法适应增多的特征量，测试集的准确率也就略有下降。

### 5.3.2 有效率

本实验取一次输入的时间片数量为 48，这是在输入不同时间片的实验中带来最优效果的时间片。该实验探究不同的数据有效率对以训练模型效果的影响，具体来说，就是使用完好的数据训练模型，而在测试时输入特征的每一个值都以一定概率置为 0（无效）。实验结果汇总于表 3中。

本小组绘制了测试准确率随有效率变化曲线，如图 12。随着有效率下降，输入特征趋向于 0，输出也趋向于 $model(0)$ 对应的分类标签在测试数据中所占的比例，其中 $model$ 表示具体使用的模型。因此当有效率较低时，达到最优效果的模型很稳定，测试集准确率也渐次趋向于有效率为 0 对应的准确率。

**数据集的不平衡性** 对于平衡的数据集来说，$model(0)$ 对应的分类标签应当占据 $\frac{1}{n}$，其中 $n$ 表示类别数量。在 Motion Sense 数据集中 $\frac{1}{n} = 16.7\%$，而 $model(0) = 31.7\%$，表示此分类标签数据占据了总量的 $\frac{1}{3}$，体现了数据的不平衡性。

| 有效率 | 有效特征数 | 最优模型 | 测试集准确率 |
|---|---|---|---|
| 0.008 | 4.6 | LightGBM | 31.7% |
| 0.02 | 11.52 | LightGBM | 31.7% |
| 0.04 | 23.04 | LightGBM | 31.8% |
| 0.06 | 34.56 | LightGBM | 32.0% |
| 0.1 | 57.6 | LightGBM | 33.2% |
| 0.2 | 115.2 | LightGBM | 44.8% |
| 0.4 | 230.4 | LightGBM | 82.0% |
| 0.8 | 460.8 | LightGBM（Extremely Randomized Trees） | 96.9% |
| 1.0 | 576 | LightGBM（Extremely Randomized Trees） | 97.8% |

表 3: 不同的有效率对效果的影响

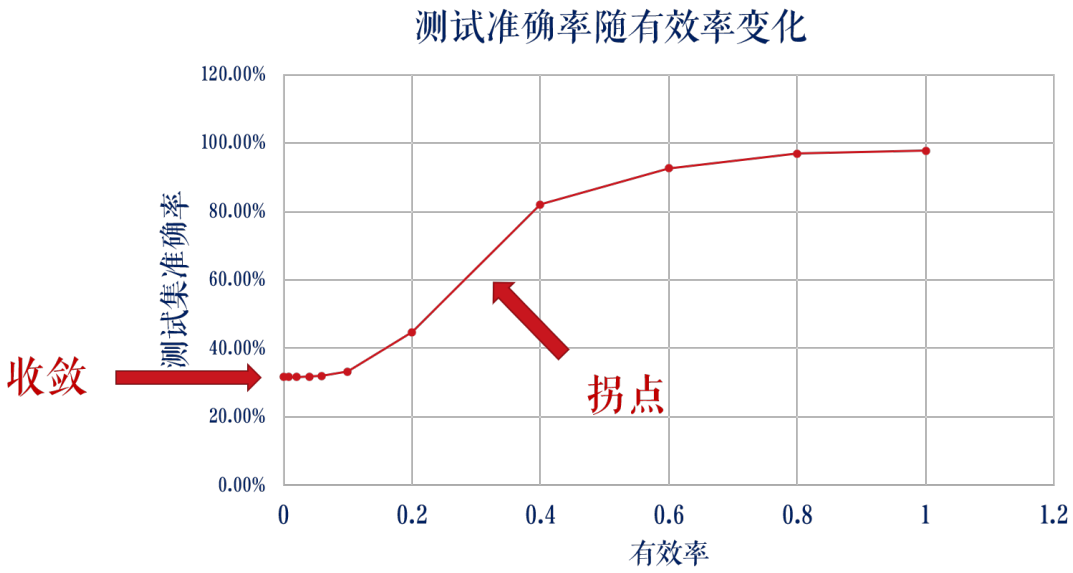



图 12: 测试准确率随有效率变化曲线

**拐点** 实验结果验证了”有效率越高，模型的性能越好“的假设。对于 Motion Sense 数据集来说，测试集准去率下降的拐点出现在有效率为 0.2-0,4 之间。

**数据集中性：猜想** 准确率下降的拐点出现在 0.2 到 0.4 之间，意味着，当输入数据在时间维度和传感器维度被掩盖掉大部分信息后，依然可以达到较好的效果。于是我们猜想，”分散的数据优于集中的数据”，我们也设计实验尝试验证了这一猜想，实验结果汇总与表 4。

**数据集中性：结论** 实验结果显示出了不同的结论。分散的数据在使用完整数据训练的模型上的效果大打折扣，在使用分散数据训练的模型上的测试集准确率略低于集中数据。值得注意的是，训练数据的准确率有较大的下降，这说明分散数据可以达到的性能还有较大的提升空间。比较遗憾的是，由于项目时间有限，我们无法再深入探究分散数据和集中数据在其他场景下的优劣，只能说明**在特征数量为 144，有效率为 0.25 时分散数据并不优于集中数据**。

| 集中性 | 时间点 | 训练有效率 | 测试有效率 | 最优模型 | 测试准确率 | 验证准确率 |
|---|---|---|---|---|---|---|
| 集中 | 12 | 1.0 | 1.0 | XGBoost | 94.7% | 97.5% |
| 分散 | 48 | 1.0 | 0.25 | CatBoost | 51.3% | 97.7% |
| | 48 | 0.25 | 0.25 | Weighted Ensemble | 93.6% | 93.9% |

表 4: 数据集中性

## 5.4 传感器屏蔽

Motion Sense 数据集使用加速度计和陀螺仪进行运动数据提取，每一种物理传感器都能得到两种类型的数据，故而数据集中包含四种类型数据：1. 位姿数据、2. 重力方向、3. 旋转方向、4. 加速度。该使用中的屏蔽的“传感器”具体指这四个种类的数据。具体而言，每一组实验中，在测试时将要屏蔽的种类的数据置为 0。实验结果汇总于表 5。

| 屏蔽传感器 | 模型 | 测试准确率 | 推理时间 |
|---|---|---|---|
| Attitude（位姿） | LightGBM（最优） | 97.3% | 36.4 |
| Gravity（重力） | LightGBM（最优） | 95.4% | 41.1 |
| Rotation（旋转） | Random Forest | 92.2% | 4.91 |
| | *LightGBM（较优）* | *88.2%* | *38.7* |
| Acceleration（加速度） | K Neighbor | 89% | 1218.4 |
| | *LightGBM（较差）* | *65.0%* | *43.3* |

表 5: 屏蔽传感器对效果的影响

**传感器重要性比较** 屏蔽掉越重要的传感器，测试准确率下降幅度越大。从表 5中的数据可以看出，传感器的重要性排序为**位姿 < 重力 < 旋转 < 加速度**。此外，实验中还发现，***LightGBM* 模型在屏蔽位姿、重力、旋转速度以后效果不错，而在屏蔽加速度数据时效果较差，是因为 *LightGBM* 模型可以很好地提取加速度数据特征**，验证此结论的实验数据如表 6所示。

| 屏蔽传感器 | 模型 | 测试集准确率 | 验证集准确率 |
|---|---|---|---|
| Acceleration（加速度） | K Neighbor | 89% | 95.3% |
| | Extremely Randomized Trees | 87.0% | 97.1% |
| | Random Forest | 72.3% | 97.2% |
| | Weighted Ensemble | 71.7% | 98.5% |
| | CatBoost | 68.7% | 97.7% |
| | **LightGBM** | 65% | 97.9% |
| | XGBoost | 51.6% | 97.6% |

表 6: 屏蔽加速度数据时不同模型的效果

**“加速度”是数据的主要成分** 屏蔽加速度传感器以后，各个模型的预测准确率如表 6所示。当屏蔽了加速度数据后，原先取得最优效果的 LightGBM 模型反而效果比较差，该实验说明了 LightGBM 取得较好效果的原因是它能更好地提取加速度数据。

# 6 结论

本项目对 CNN+LSTM 模型进行了复现，并比较 CNN 和 LSTM 模型的时序表达能力。实现并尝试了多种不同的机器学习模型的效果，最高达到 **97.8%** 的准确率。探究 Motion Sense 运动探测数据集的数据内在特性。达到的结论如下：

**模型对比** **1.** 时序数据的表达能力：LSTM 远大于 CNN。**2.** 参数越大，表达能力不一定随之增大，数据特性的影响甚至远超模型参数量的影响。

**数据内在特性探究** **1.** 更多输入特征带来更丰富的输入信息，一般会带来更好的预测效果，但也需要表达能力更强的模型来提取。**2.** 在特定场景下，集中数据的内在联系更紧密，效果优于分散数据。**3.** 传感器重要性：加速度 > 旋转速度 > 重力方向 > 位姿。**4.** 占据主导的数据成分决定了相匹配的模型。

# 7 总结

本次课程项目在运动检测领域进行了研究实践和创新。项目的内容可以总结如下。本小组查阅了运动检测领域的相关文献，复现了 [2] 的神经网络结构。之后，对模型进行创新和改进，将 GBDT，LightGBM，RondomForest，KNN 以及集成学习模型应用于运动检测领域，提升了模型的准确度和分类效率。最后，进一步探究 MotionSense 数据集中的内在规律，通过过更改时间片长度，修改实验数据有效率以及传感器屏蔽等方法，分析实验结果，得到了数据量有效性对学习结果的影响程度以及不同类型传感器间的数据共享程度等相关结论。

综上所述，本小组在模型改进和数据规律挖掘方面做出了相应的工作，并获得了相关的实验结果。这为我们日后在运动检测领域的进一步学习和探索打下了基础。在本次课程项目的实践过程中，我们小组成员通力合作，面对困难及时查找资料，乐于探索。既收获了相关领域的知识，也锻炼了自身的学习能力。最后，感谢在本次项目实践过程中，顾磊磊老师，盛斌老师以及助教们的悉心指导和解答。

# 参考文献

[1] Malekzadeh, Mohammad, et al. ”Mobile sensor data anonymization.” Proceedings of the international conference on internet of things design and implementation. 2019.

[2] Kanjo, Eiman, Eman MG Younis, and Chee Siang Ang. ”Deep learning analysis of mobile physiological, environmental and location sensor data for emotion detection.” Information Fusion 49 (2019): 46-56.

[3] Friedman, Jerome H. ”Greedy function approximation: a gradient boosting machine.” Annals of statistics (2001): 1189-1232.

[4] Caruana R, Niculescu-Mizil A, Crew G, et al. Ensemble selection from libraries of models[C]//Proceedings of the twenty-first international conference on Machine learning. 2004: 18.