通过机器学习预测电网平均总负荷

该研究基于比利时 Elia 电网公司的公开数据，该数据给出了过去几年以 15 分钟为间隔测量的 Elia 电网的总负荷。此处将使用机器学习方法提前一天预测 Elia 的平均总负荷。研究表明，在对负荷时间序列数据进行预处理后，使用机器学习方法可以产生低预测误差。以上成果可用于实际的生产生活中，电力公司可以使用预测值来产生足够的发电量，以避免电网中断和电力损失，并根据未来的负荷构建动态定价方案。

1 引言：

对于输电运营商来说，电力系统的管理是一项复杂的任务，并且在很大程度上依赖于对未来电力需求的了解。可以准确预测负荷的模型对于能源生产至关重要，因为根据预测的负荷可以确定应该运行哪些设备以满足需求，更好地满足机组组合优化。若不能产生足够的能量会导致电网故障，若供过于求会导致能源和资源的浪费。随着去中心化电力市场的出现，必须根据当前需求制定准确的定价协议，因此了解未来需求至关重要。

具体来说，该项预测任务是提前一天预测 Elia 电网的平均总负荷。总负荷包括 Elia 电网上的所有电力负荷，包括底层配电网络以及电力损耗。用于出口和储能的电力将从该值中扣除，因此总负荷值是对 Elia 电网和底层网络实际负载的估计。 每天以 15 分钟为间隔测量总负荷值，每天产生 96 个值。 特定一天的平均总负载是当天 96 个总负载值的平均值。我们的目的是在每天之前以尽可能小的误差预测这些值。为此，我们将构建基于统计和机器学习的方法，预测 2017 年各天的平均总负载并测试其性能。

文章阐述结构如下。

研究动机部分：详细说明准确预测电力负荷的重要性。

相关工作部分：描述过去在预测电力负荷方面所做的研究并总结迄今为止已获得的结果。

统计预处理部分：描述我们面临的预测问题，并提供了用于负荷时间序列数据的 scaling 缩放和 detrending 去趋势方法。

算法部分：描述几种不同的机器学习算法，以及如何将它们与预处理步骤结合从而形成预测模型。

实现部分：将描述用于构建和测试我们模型的硬件架构和关键软件库。

实验结果部分：应用我们的方法来预测 Elia 电网的平均总负载，并根据基准衡量机器学习方法的性能。

未来的方向：提出可以建立在我们工作基础上的扩展和新方向。

最后的结论：简要总结我们的研究成果。

2 研究动机

负荷预测对于电力公司来说是一项至关重要的任务，其对于确定公司应提供的电量至关重要。 无法满足需求可能会导致网络组件过载和大范围的电网中断，这两种情况都会造成经济损失并损害电力公司的声誉。 准确估计未来网络流量可以让供应商规划发电、开发电网基础设施并响应变化的频率，以避免这些结果并提高整体网络可靠性。

准确负荷预测的一个经济效益是能够根据总需求对电力进行定价。能源市场的去规范化和激烈竞争使得输电运营商能够以与其他能源供应商竞争的价格定价电力变得至关重要。 这些价格会随着电力需求而波动，高峰期的特点是电价高，非高峰期的价格也较低。 动态确定这些时间段何时发生以及这些时间段内的电力需求可以帮助制定基于需求的具有竞争力和公平性的定价方案。

过去，人们采用各种预测方法来预测未来负荷，包括计量经济学方法、基于回归的模型和统计学习算法。这些方法取得了不同程度的成功，但仍有很大的改进空间。 我们的目标是将机器学习算法应用于负荷预测问题。此外，我们将使用时间序列分析方法对数据集进行缩放和去趋势化，为学习算法做好准备。由于上述原因，会得到更准确的负荷预测，并证明对电力公司很有价值。

3 相关工作

Taylor 等人在（[1]）中发现电力负荷时间序列中存在季节性模式，我们的研究将对 Elia 数据集中的季节性模式进行统计测试，并将该信息用于其中一种机器学习方法，研究证明该机器学习方法可以减少预测

误差。

Chen 等人在（[2]）中也采用了一种机器学习方法来预测未来的电力负荷，它使用支持向量机（SVM）来预测未来的电力负荷。他们最高性能的 SVM 通过从过去几天的需求中形成特征向量来合并时间序列信息，并使用它们来预测未来几天的负荷。 在我们的研究中，我们将像本研究一样使用 SVM 进行预测，但将从日内电力负荷而不是从多天数据中构建特征向量。

最后，我们也包含了有 Ahmed 在 ([3]) 中的工作，它凭经验证明了在应用机器学习算法进行预测之前进行统计预处理时间序列数据的好处。 证明了数据归一化、对数缩放、趋势去除和去季节性可以提高判别学习算法的性能，例如 K 最近邻和支持向量机。 在我们的方法中，我们将在应用机器学习方法之前对数据进行对数缩放并去除趋势分量，并将季节性信息用于基于聚类的学习算法。

4 统计学处理

首先介绍一些将在整篇论文中使用的符号：

xTrain - 按时间顺序组成 2008-2016 年负荷时间序列的向量序列。 每个向量都是 96 维的，因为每十五分钟测量一次电网上的总负荷。

yTrain - xTrain 中样本对应的第二天平均总负荷。平均总负荷是指第二天所有 96 个负荷值的平均值。

xTest , yTest - 除 2017 年外，其他的与训练数据集相同。

训练和测试集中的最后一个样本需要提取下一年的负荷值，这是作为预处理的一部分完成的。 我们在图 1 中绘制了相应时间段内的 yTrain 值。从图中可以看出，除了负荷的大数值外，似乎还有显着的趋势和季节性分量。 以下预处理步骤将解决这些问题，使数据更易于学习。

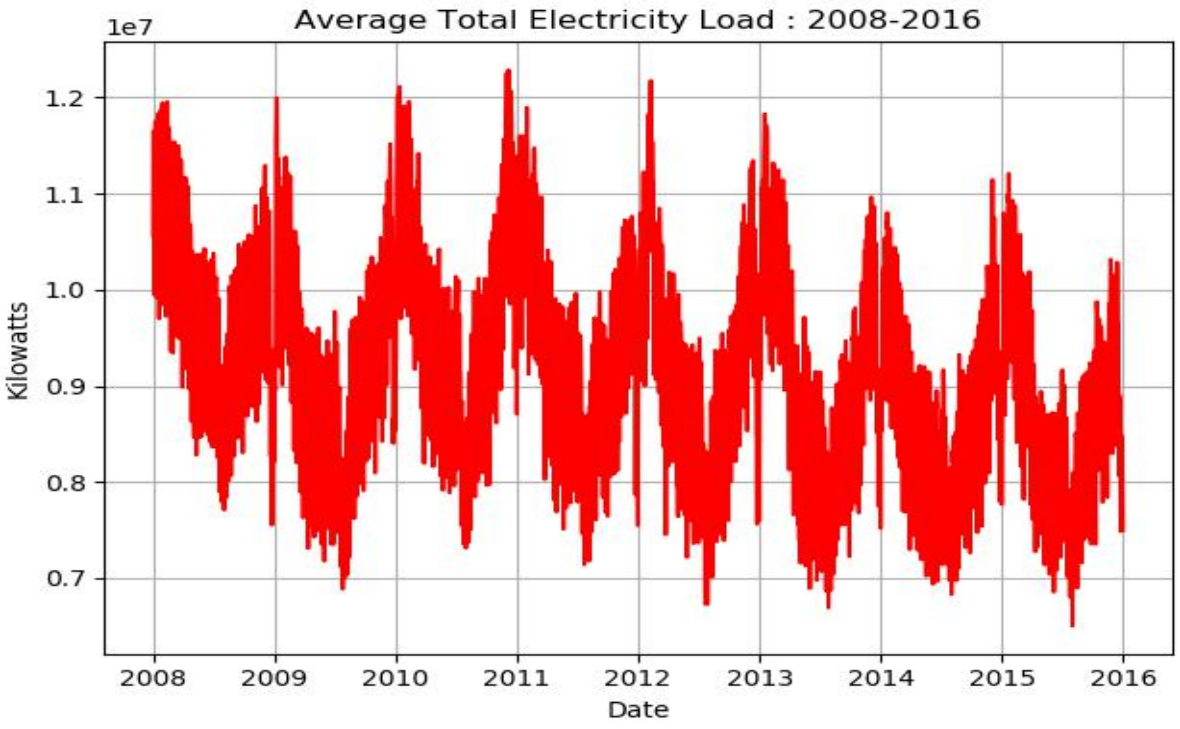

图 1 2008-2017 平均总负荷图

4.1 数据预处理

预处理阶段有 2 个步骤：

（1）对数缩放

时间序列包含非常大的数值，这会阻碍学习器有效学习的能力并使问题计算密集。为解决以上问题，

在学习之前，xTrain、yTrain 和 xTest 中的值将通过自然对数进行缩放。

（2）趋势消除

消除目标值中的趋势可以通过消除时间概念来提高预测准确度。 为此，我们首先通过对 yTrain 数据集执行最小二乘线性回归来估计趋势。每个值对应于特定的一天，因此我们可以将 yTrain 中的值视为按时间顺序排列的值序列。回归的自变量是从序列开始算起的天数，因变量是该特定日期的 yTrain 值。一旦计算出回归线，我们从 yTrain 中的每个值中减去回归预测值，从而获得形成去趋势时间序列的残差。xTrain 值不受影响。

4.2. 季节性分析

根据之前的研究和观察，图 1 中 yTrain 的图，我们认为时间序列包含显着的季节性成分。为了验证该猜想，我们继续对季节性行为进行统计测试。

首先，我们在图 2 中显示 yTrain 的滞后相关图，其中绘制了 yTrain 的值与其前一天的值。 这张图使我们能够确定基础时间序列是否是随机的。如果它是随机的，那么滞后图不应显示任何模式或有意义的子结构。 然而，根据该图，我们可以观察到一个显着的线性趋势，这表明可能是由季节性引起的某些潜在的模式。

现在我们有了季节性成分的证据，我们将使用相关图来确定季节的周期。相关图根据时间滞后绘制自相关因子，其中自相关因子是在某个给定时间滞后下 yTrain 与自身的协方差除以 yTrain 的方差。 在数学上，对于时间滞后为 k 的 yTrain 的具有较小偏差的自相关因子 rk 的估计可以表示为：

$$r_k = \frac{1}{n-k}\left(\frac{\sum_{j=1}^{n-k}(Y_{train}^{j}-\mu_Y)(Y_{train}^{j+k}-\mu_Y)}{\sum_{j=1}^{n}(Y_{train}^{j}-\mu_Y)(Y_{train}^{j}-\mu_Y)}\right) = \frac{1}{\sigma^2(n-k)}\left(\sum_{j=1}^{n-k}(Y_{train}^{j}-\mu_Y)(Y_{train}^{j+k}-\mu_Y)\right)$$

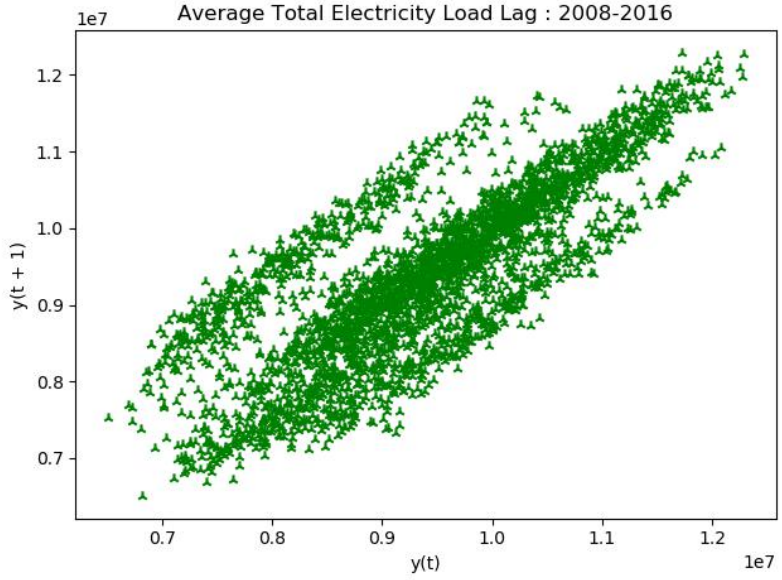

图 2 2008-2016 年平均总负载的滞后图

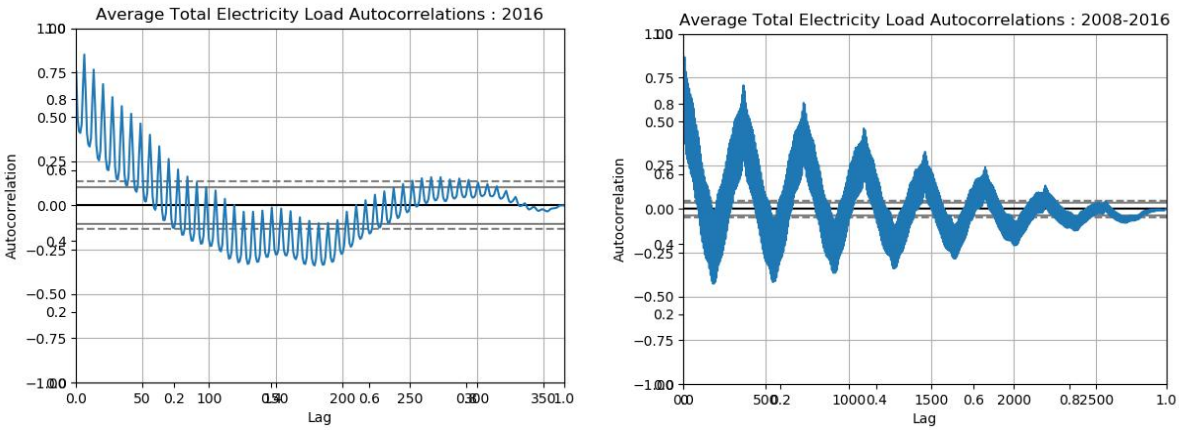

图 3 平均总负载在不同时间滞后的自相关因子的相关图

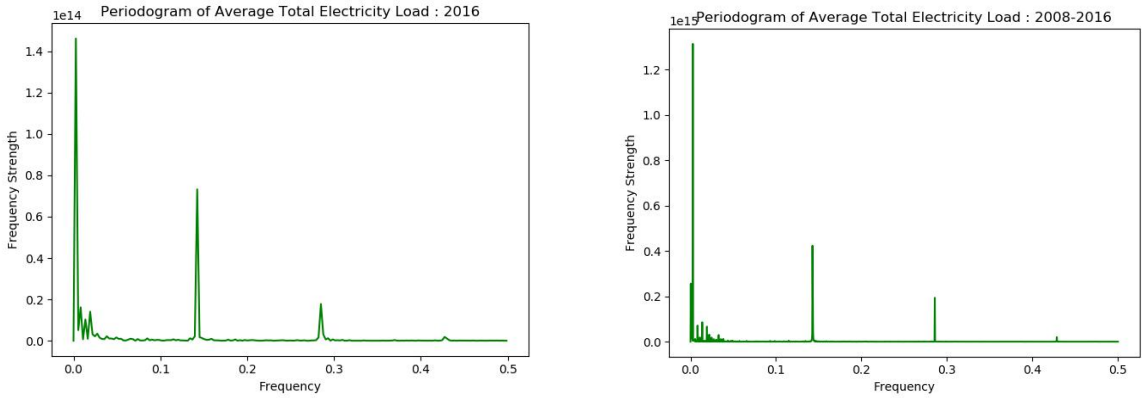

图 4 用于频域分析的平均总负载周期图

我们让 n = |yTrain| ，并将 yTrain 的平均值定义为 $u_Y = \frac{1}{n}\sum_{i=1}^{n} Y^i_{train}$。假设值 yTrain 是独立同分布的。自相关因子的均值和方差分别为 -1/n 和 1/n 。 这些可用于查找 95% 和 99% 置信区间，它们在相关图上显示为水平线。

从相关图中我们可以观察到，2008 - 2016 年图上的滞后一年和 2016 年图上的滞后一周都产生超过 99% 置信区间的自相关因子。 此外，我们构建了周期图，可用于识别频域中时间序列的重要周期。 通过检查周期图，我们观察到每周和每年频率的峰值，这表明这些时间滞后的时间序列有很强的相关性。 根据这些统计检验的结果，我们可以得出结论，时间序列很可能包含显着的季节性成分。 我们将使用每周和每年作为基于聚类的机器学习算法的季节性周期，这将在下一节中描述。

5 算法

5.1 支持向量回归 SVR

第一个基于机器学习的预测方法使用支持向量机回归 (SVR)。 传统上支持向量机用于二元分类任务，并在高维度上找到欧几里得空间中的最佳超平面，将数据分为两类。 该算法已经针对回归任务进行了修改，因此我们可以将其应用于当前的问题。 具体来说，SVR 将解决以下优化问题，以原始形式呈现如下：

$$Minimize\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i}^{n}(\xi_i + \xi_i^*)s.t.\forall i, 1 \leqslant i \leqslant n,$$

$$w^T x_i + b - y_i \leqslant \varepsilon + \xi_i$$

$$-w^T x_i - b + y_i \leqslant \varepsilon + \xi_i^*$$

$$\xi_i, \xi_i^* \geqslant 0$$

由于已知SVR以在高维度下进行判别学习任务而擅长，因此它为我们的预测任务提供有价值的方法。然而，鉴于上述方程式，我们仅限于做原始矢量空间中的线性分类，这可能是次优级别的学习。 为了克服这种限制，我们获取了原始方程的对偶式：

$$Maximize -\frac{1}{2}\sum_{i,j=1}^{n}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)x_i^T x_j \ and$$

$$\sum_{i=1}^{n} -\varepsilon(\alpha_i + \alpha_i^*) + y_i(\alpha_i - \alpha_i^*) \ s.t.$$

$$\sum_{i=1}^{n}(\alpha_i - \alpha_i^*) = 0$$

$$\alpha_i, \alpha_i^* \in [0, C)\forall i, 1 \leqslant i \leqslant n$$

6

Smola 在 [11] 中描述了这两种方程。 我们可以用一个核函数 k 代替对偶公式中的内积项，k 度量了高维向量空间中的内积的相似性，并允许 SVR 学习关于原始向量空间的非线性模式。 这可能会导致更准确的预测器，这将在实验中进一步研究。 则回归函数 f 为：

$$f(x) = \sum_{i=1}^{n}(\alpha_i - \alpha_i^*) * k(x, x_i) + b$$

现在定义基于 SVR 的预测算法。该算法的输入由核函数 k 和误分类惩罚系数 C 以及训练和测试数据集组成。 该算法将缩放和去趋势步骤应用于 xTrain 和 yTrain，如预处理部分所述，以获得训练集。 然后将 SVR 与内核 k 和惩罚 C 与 训练集一起使用以获得预测器 f。 然后对于每个样本 x ∈ xTest，算法对进行缩放并应用 f 以获得其对应的预测值。 然后将趋势分量加回到该预测值上，通过逆自然对数重新缩放以获得最终预测值。

5.2 聚类 Clustering

第二个机器学习方法是聚类，并使用预处理部分中检验过的季节性信息。 我们将季节长度视为样本的唯一类别的数量，其中每个类别对应于任何给定季节的偏移量。在聚类方面，我们希望为这些类中的每一个构建一个集群，以便集群中的样本表现出相似性，同时与其他集群中的样本不同。 为了实现这一点，我们可以对样本进行聚类。然后给出一个测试样本，根据与它最相似的那些样本来预测它的值，它们是与测试样本具有相同季节性偏移的样本。

我们将尝试两种不同的聚类算法。 第一种是自底向上类型的层次聚类，称为凝聚法层次聚类，它是一种无监督学习算法，给定一组向量和聚类数 k，找到 k 个集群 S = Φ1, ..., Φk 使得每个向量都恰好属于集群之一，并且集群间距离最大化。集群之间的距离将使用欧几里德距离和离差平方和法(ward)计算，其目标是合并集群以最小化集群内的总方差。

第二种聚类方法是 K-Means。K-Means 算法将使用欧几里德距离度量将一组向量聚类到 k 个簇中，以便最小化簇内平方和。。

现定义基于聚类的算法。 算法的输入包括季节长度 s、聚类算法的选择（K-Means 或 凝聚层次聚类），

以及训练和测试数据集。首先对数缩放和去趋势 xTrain 和 yTrain 以获得训练集。 然后在训练集上运行聚类簇数为 s 的两种聚类算法，以获得一组集群 S = ϕ1, ..., ϕs ，各自具有相应的簇心 μ1, ..., μs 。 然后对于每个样本 x ∈ xTest，应用对数缩放步骤来获得 x0。 找到簇心为 μ* 的簇 ϕ* ∈ S，使得 μ* 和 x0 之间的欧几里德距离最小，可以表示如下：

$$\phi^* = argmin_{\phi_i \in S} d(\mu_i, x')$$

其预测值为 y0 。它是集群每个成员的预测值乘以它们和 x0 之间的欧几里德距离的加权和，然后由通过它们和 x0 之间的欧几里德距离之和进行标准化。可以将该预测表示为：

$$y' = \frac{\sum_{\{x_t \in \phi^*\}} d(x_t, x') * y_t}{\sum_{\{x_t \in \phi^*\}} d(x_t, x')}$$

最后将趋势添加到 y0 并用逆自然对数重新缩放以获得预测 y。

5.3 神经网络 Neural Networks

第三种学习方法是神经网络，多年来神经网络已被用于判别学习任务，并取得了大量成功。前馈神经网络是一个无环有向图结构 G = (V, E)，其中节点 V 是神经元，其通过一组有向边 E 相互连接。 有一组输入神经元接收一个样本并将样本向量的值输出到后续的隐层神经元。这些隐层神经元采用其输入的线性组合，可能会添加一个偏置值，然后将激活函数应用于该计算的总和，并将获得的值沿传出边缘输出到其他神经元。以这种方式，值从输入层通过网络传播，直到它们到达输出层，在输出层，误差函数被应用于输出层神经元的输出值。然后通过著名的反向传播算法更新网络，该算法尝试通过调整网络权重来最小化获得的误差。我们使用的神经网络的机制如下：

神经元的输入 zj 和输出 yj 可以表示为：

$$z_j = \sum_{i|e=(i,j) \in E} y_i w_{ij}$$

$$y_j = \frac{1}{1 + exp(-z_j)}$$

预测值 yj 与真实值 ˆy 的平方损失是：

$$\zeta(y_j, \hat{y}) = \frac{1}{2}(y_j - \hat{y})^2$$

定义一个误差项 εj，这将使后续的梯度表达式更简单。

输出神经元 j 的误差项 εj 是：

$$\varepsilon_j = (y_j - \hat{y})(\frac{exp(-z_j) - 1}{(1 + exp(-z_j))^2})$$

输入神经元 j 的项 εj 是：

$$\varepsilon_j = (\sum_{k|(j,k) \in E} \varepsilon_k)(\frac{exp(-z_j) - 1}{(1 + exp(-z_j))^2})$$

连接权重到输出神经元的梯度为：

$$\frac{\partial \zeta}{\partial w_{ij}} = \frac{\partial \zeta}{\partial y_j}\frac{\partial y_j}{\partial z_j}\frac{\partial z_j}{\partial w_{ij}}$$
$$= (y_j - \hat{y})(\frac{exp(-z_j) - 1}{(1 + exp(-z_j))^2})(y_i) = \varepsilon_j y_i$$

同样，输入神经元的连接权重梯度为：

$$\frac{\partial \zeta}{\partial w_{ij}} = \frac{\partial \zeta}{\partial y_j}\frac{\partial y_j}{\partial z_j}\frac{\partial z_j}{\partial w_{ij}}$$
$$= (\sum\nolimits_{k|(j,k)\in E} \varepsilon_k)(\frac{exp(-z_j) - 1}{(1 + exp(-z_j))^2})(y_i) = \varepsilon_j y_i$$

神经网络将使用随机梯度下降 (SGD) 进行训练，它从样本的子集而不是整个数据集估计梯度。以这种方式更新权重可以加快收敛速度。权重更新规则可以简明地写为：

$$w_{ij}^{t+1} = w_{ij}^{t} - \lambda(\sum_{\alpha=1}^{N} \varepsilon_j^{\alpha} y_i^{\alpha})$$

其中 SGD 在更新前观察 N 个样本，εαj 和 yαi 是样本 xα 上的误差和神经元输出条件。 值 t 代表当前迭代次数，值 λ 是控制权重更新速度的学习率。

最后，与其他机器学习算法不同，我们将使用主成分分析 (PCA) 来降低特征向量的维数。这样做的原因是训练集包含大约 3000 多个样本，由于参数数量的原因，这对于训练具有 96 个输入神经元的网络来说太少了。 因此，我们使用 PCA 将样本投影到较低维度的子空间中，同时保持最大的方差量，这会保留区分样本所需的信息，也减少了输入神经元的个数。

基于神经网络的预测算法的工作原理如下。算法的输入由训练和测试数据集、隐藏神经元数量的正整数 h 和 PCA 的维度 d 组成（激活函数和误差函数是固定的，并且有一个隐藏层，正如我们观察到的交替执行更糟糕）。首先对数缩放和去趋势 yTrain 以获得 Y'train 。对数缩放并将目标维度为 d 的 PCA 应用到 xTrain 以获得 X' train，保留用于 PCA 的投影矩阵 W。然后构建一个具有 d 个神经元的输入层、一个全连接的 h 个神经元的隐藏层和一个具有来自所有隐藏神经元的输入边的单个输出神经元的神经网络。神经元激活函数是 sigmoid，损失是平方损失，如前所述。对于每个示例 x ∈ xTest，对 x 进行对数缩放以获得 x ^，然后通过使用公式 x0 = W ^x 进行投影获得 x0。将神经网络应用于 x0 得到输出 y0 ，然后加入趋势分量并用逆自然对数重新缩放以获得预测值 y。

5.4 高斯过程回归 Gaussian Process Regression

第四种学习方法将使用高斯过程回归 (GPR)，这是一种主要用于回归任务的判别学习算法。设 X = x1, ..., xn 为具有相应目标值 y = y1, ..., yn 的训练样例。 然后 GPR 将尝试学习以下形式的模型：

$$y(x) = f(x) + \varepsilon$$

其中 f(x) 是回归分量， ε 是均值为零且方差为 σ 2n 的高斯噪声项。

GPR 预测算法如下。 算法的输入由协方差函数 k 以及训练和测试数据集组成。 该算法对 xTrain 和 yTrain 应用对数缩放和去趋势操作，以获得 X0 训练和 Y0 训练。 然后在 X0 训练和 Y0 训练上使用 GPR 计算预测变量 f，协方差度量为 k。 然后对于每个示例 x ∈ xTest 应用对数归一化以获得 x0 并计算 y0 = f(x0)。 然后将趋势分量加回到 y0 并应用逆自然对数以获得预测 y。

5.5 其他机器学习算法

除以上四种方法外，我们还尝试了其他机器学习算法来探索其在电力负荷预测上的能力，如逻辑回归、决策树、随机森林、XGboost，以及长短时记忆网络、门控递归单元神经网络等深度学习算法。该部分内容不作为研究的主要内容，后续可进行持续探索。

6 实施

该项目是用 Python 实现的，大量使用了 scikit-learn 机器学习库、Numpy 数值计算库、Pandas 数据分析库、Pybrain 机器学习库、keras 深度学习库。 系统架构如下：

- Win10 家庭版 64 位操作系统
- CPU 处理器（Intel(R)Core(TM)i5-6200U）
- 8GB 内存

7 实验结果

将四种基于机器学习的算法与计量经济学模型一起应用于预测 2017 年平均总负荷的任务中。对于每种算法，我们尝试了几种不同的参数，并在单个图上显示每种配置的结果。 图 5、6、7、8 分别显示了 SVR、聚类、GPR 和神经网络结果。

我们选择根据归一化均方根误差 (NRMSE) 来衡量性能，因为它是经过缩放的，并且允许在算法之间进行公平的比较。 如果让 Ypred 和 Ytest 分别为 n 个预测值和对应的测试值，则 NRMSE 的函数可以表示为：

$$NRMSE(Y_{pred}, Y_{test}) = \frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_{pred}^{i} - Y_{test}^{i})^2}}{Y_{pred}^{MAX} - Y_{pred}^{MIN}}$$

我们测量了所有算法及其各种配置的 NRMSE，并在表 1 中显示了结果和参数选择。

我们立即注意到，最低的 NRMSE 值 0.10558 是通过 GPR 实现的。 GPR 超越聚类、SVR 和神经网络算法的事实证明了 GPR 在数据集不太大（例如，最多几千个示例和几十个维度）时在判别回归学习任务可实现高准确率。然而，除了预处理步骤之外，GPR 模型中没有使用有关季节性的信息，尽管之前的研究和我们的初步分析表明负荷时间序列包含显着的季节性成分。这表明季节性信息可能不是实现高准确率预测平均总负荷所必需的。但我们发现归一化和去趋势的预处理步骤对于机器学习方法的成功至关重要，因此表明时间序列分析的思想与机器学习结合使用可以开发强大的预测模型。

8 未来方向

基于我们的结果，我们有强有力的证据表明基于机器学习的算法能够实现负荷预测问题的高准确率，因此我们相信进一步探索这一领域将是有益的。

一个直接的方向是尝试其他判别机器学习算法，如贝叶斯神经网络、k-最近邻、回归方法（岭、逻辑、贝叶斯）和决策树，并根据本文中的结果衡量它们的性能。

由于我们的算法通常有很多参数（SVR 有核函数和误差惩罚，GPR 有协方差函数），另一个方向可能是尝试改变参数，看看另一个选择是否会产生更低的错误率。

此外，多种算法可以与集成方法一起使用，历史上已经证明在某些情况下它们优于集成的组成算法。

此外，研究时间序列分析中的方法以及如何将它们与机器学习结合使用似乎也很有价值。 在这一点上，有多种时间序列方法可用于更改和转换数据集，从而可以更深入地了解和分析负责生成数据的底层过程。一些方法包括了如噪声检测和去除、去季节性（尽管我们发现加法分解无效）和平滑（卡尔曼滤波、拉普拉斯算子）。 将这些技术与学习算法一起使用可能会产生强大而准确的预测器，这些预测器的性能可能优于使用纯计量经济学或机器学习方法。

最后，将本文中使用的技术应用到从其他电力公司处收集其他电力负荷时间序列数据应用会很有用。我们认为，电力负荷时间序列具有某些共同点，例如季节性影响和趋势，并且本文中提出的预测方法至少会在此类数据集上取得一定程度的成功。天气数据也可以被收集并用于预测，其他研究人员已经证明这与

电力使用密切相关。可以将获得的结果都可以与电力公司使用的结果进行比较，看看它们是否比工业中使用的结果更准确。还可以与现实世界的标准方法进行比较，以确定电力公司目前使用的预测方法是否可以改进。

9 结论

根据我们的经验结果，我们已经成功地证明了机器学习技术可以产生准确的预测器，用于提前一天预测 Elia 电网的平均总负荷。但季节性信息并没有帮助提高使用高斯过程回归的最准确分类器的性能，尽管它是基于聚类的方法的组成部分，也实现了相当低的错误率。然而，我们发现对数缩放和去趋势的负荷时间序列显着提高了基于学习的模型的准确性。总的来说，我们相信我们的方法清楚地证明了利用时间序列和学习方法的价值，我们推测这将是未来负荷预测工作中不可或缺的一部分。该问题仍然是一个活跃的研究领域，远未解决，希望电力运营商继续寻求改进的预测方法，因为负荷预测具有重要的实用价值。

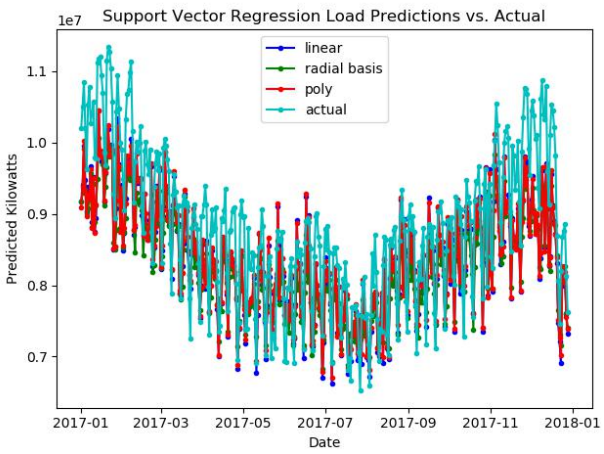

图 5 SVR 预测 2017 年平均负荷

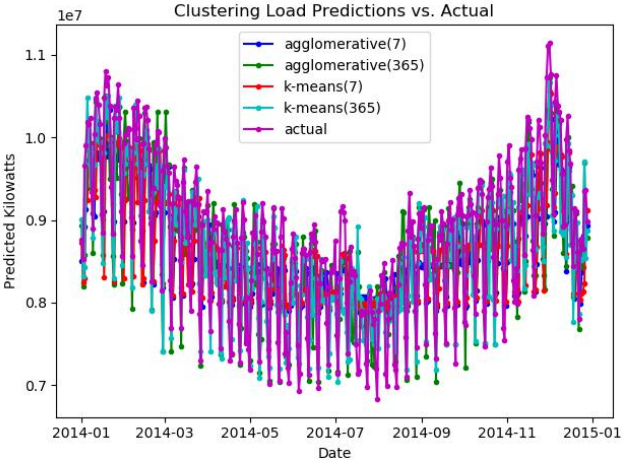

图 6 Clustering 预测 2017 年平均负荷

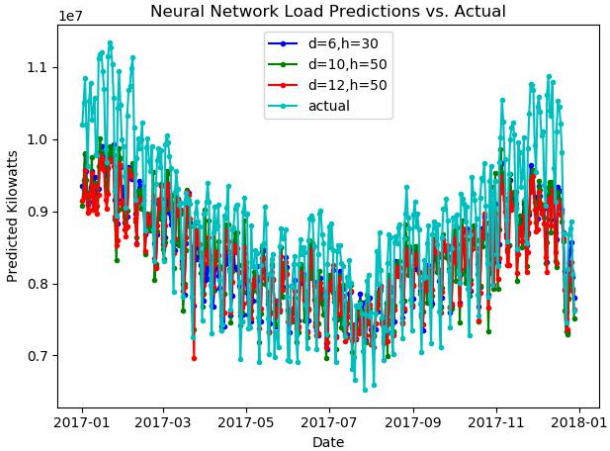

图 7 MLP 预测 2017 年平均负荷

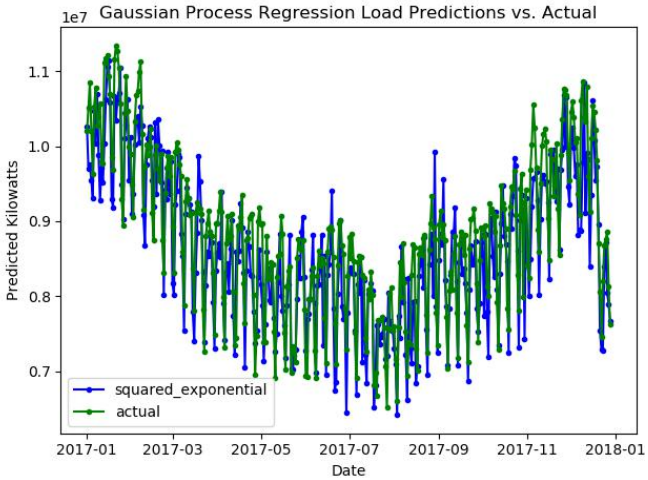

图 8 GPR 预测 2017 年平均负荷

表 1 算法比较

Algorithm	Parameters	NMSE
SVR	kernel linear	0.14325
SVR	kernel radial basis	0.15976
SVR	kernel poly	0.14209
GPR	RBF	0.10558

Clustering	agglomerative(7)	0.18087
Clustering	agglomerative(365)	0.15579
Clustering	k-means(7)	0.18244
Clustering	k-means(365)	0.15236
neural network	h=30,d=6,(40 epochs, λ =0.015)	0.15512
neural network	h=50,d=10,(40 epochs, λ =0.015)	0.15117
neural network	h=50,d=12,(40 epochs,λ=0.015)	0.16168

参考文献

【1】James W Taylor, Lilian M De Menezes, and Patrick E McSharry. A comparison of univariate methods for forecasting electricity demand up to a day ahead. International Journal of Forecasting, 22(1):1–16, 2006. 3

【2】Bo-Juen Chen, Ming-Wei Chang, and Chih-Jen Lin. Load forecasting using support vector machines: A study on eunite competition 2001. Power Systems, IEEE Transactions on, 19(4):1821–1830, 2004. 3

【3】Nesreen K Ahmed, Amir F Atiya, Neamat El Gayar, and Hisham El-Shishiny. An empirical comparison of machine learning models for time series forecasting. Econometric Reviews, 29(5-6):594–621, 2010. 3