应用深度学习方法预测复杂季节性行为的经验总结

Andrew T. Karl[1], James Wisnowski[1], Lambros Petropoulos[2]

[1]Adsurgo LLC，彭萨科拉，佛罗里达州[2]USAA，圣安东尼奥，得克萨斯州

摘要
近年来，深度学习方法借助媒体宣传，以及通过 Keras 等开源软件包相对容易实现的优势，日益受到关注。我们研究了常用循环神经网络在一家大型金融服务公司呼叫中心呼叫量预测中的适用性。这些时间序列极其复杂，既呈现季节性模式（包括一天中的不同时段、一周中的不同日期以及一年中的不同时间），又存在各个观测值之间的自相关。尽管我们的研究聚焦于金融服务业，但针对周期性非线性行为建模的建议适用于所有行业。我们从实践角度探讨埃尔曼（简单）、长短期记忆（LTSM）和门控循环单元（GRU）循环神经网络的参数设置与收敛准则优化。我们使用实际呼叫中心数据开展设计实验，涵盖多种不同的“技能组”（收入呼叫流），并将表现最佳的循环神经网络配置与其他现代及经典预测技术进行比较，评估指标为验证误差率。最后，我们总结深度学习方法在预测中的实用价值及其应用时需要考虑的因素。

关键词： 自回归积分滑动平均模型、时间序列

1. 引言

会员联络呼叫中心的呼叫量会因星期几、一天中的时段、节假日、业务状况及其他因素而波动。呼叫中心经理需要准确预测未来的呼叫量，以便高效管理人员配置水平。文献中对呼叫中心到达过程已有充分记录和探讨（Gans、Koole 和 Mandelbaum，2003）。在此处介绍的应用中，根据呼叫中心坐席的能力，来电可能被路由至多个不同的“技能组”（或“数据划分”）；对于每个技能组，都需要进行短期到达量预测，以预测次日或下一周的情况。

呼叫到达的每周季节性可通过多种方法有效建模，例如温特季节性平滑法（Winters，1960年）或自回归积分滑动平均模型（Box 和 Jenkins，1970）。面向从业者、介绍许多此类概念的易读参考资料在文献中已有充分记载（例如 Bisgaard 和 Kulachi（2007、2008））；推荐读物则包括 Bisgaard 和 Kulachi（2011）以及 Montgomery、Jennings 和 Kulachi（2015）。

为改进这些经典方法，“双重随机”线性混合模型（Aldor-Noiman, Feigin, & Mandelbaum, 2009）已能有效地对其他

复杂因素进行建模，具体内容见 Ibrahim、Ye、L'Ecuyer 和 Shen（2016）近期发表的综述论文。类似地，研究人员建议采用循环神经网络（RNN）作为深度学习方法，预测无线网络的呼叫量（Bianchi等人，2017）；此外，这类方法还有许多其他应用，包括预测 Uber 的乘车量（Zhu & Laptev，2017）。双重随机方法和循环神经网络方法通过纳入外生变量，在预测呼叫量时能够更灵活地对复杂的到达行为进行建模，但这种灵活性也带来了更高的计算复杂度和编程复杂度（以及更大的预测方差）。本文探讨了管理这些模型复杂度的实际问题，将模型应用于一家大型金融服务公司记录的实际呼叫量，并将其预测能力与更传统的温特斯平滑法和自回归积分滑动平均模型进行比较。

首先，我们对 Aldor-Noiman, Feigin, & Mandelbaum（2009）提出的双随机方法的计算方面进行改进，以提升呼叫中心预测的表现。双随机意味着采用两级随机化：不仅呼叫到达是随机变量，呼叫到达均值参数也是随机变量。预测利用每个数据划分特有的相关结构，同时考虑趋势、季节性、周期性行为和序列依赖。双随机模型比普通回归更复杂，因为它同时考虑日间和日内相关性。我们提出对原始方法进行改进，使其能够稳定收敛，并在需要拟合许多数据划分时表现得更加灵活。

其次，我们探讨如何使用循环神经网络对呼入呼叫量建模。传统的全连接前馈神经网络会独立处理每个数据点，而循环神经网络则按照时间顺序处理序列，并保留序列中先前数据点的信息。循环神经网络处理序列中的数据点时，会维护包含序列中先前所见信息的状态（Chollet & Allaire，2018）。这种序列内存对于自相关数据的时间序列应用很有用。在呼叫中心呼叫量的背景下，可以构建这些序列，使其对应于固定数量（例如，每30分钟一个）的时期内各日期的观测值。Bianchi等人（2017年）考察了三种不同的循环神经网络架构，用于对移动电话网络中的呼入呼叫量建模：Elman循环神经网络（ERNN）（Elman，1990）、门控循环单元（GRU）（Cho等人，2014）和长短期记忆网络（LSTM）（Hochrieter & Schmidhuber，1997），复杂程度依次递增。
这三种循环神经网络以及全连接神经网络现已可通过 R Keras软件包使用（Allaire & Chollet，2018）。用户为其中一种循环神经网络编写代码后，只需切换一个选项，就能在另外两种循环神经网络之间切换（重新整理数据后，也可以切换到全连接网络）。通过设计实验，这种方法有望为“哪种类型的（循环）神经网络最适合当前过程”这一问题提供实用答案。Bianchi, Maiorino, Kampffmeyer, Rizzi, & Jenssen（2017年）在设计实验时，先在设计空间内随机生成点，再选取错误率最低的设计；而我们采用全因子设计（将所有因素都视为分类变量，从而允许节点数这一离散连续因素呈现任意形状），然后通过剖析器，探索错误率关于神经网络设置的线性模型在设计空间内的错误率表现。与自回归积分滑动平均模型或回归（包括双重随机）时间序列建模方法不同，神经网络的预测具有随机性，这是因为其使用了随机初始化权重。除非固定软件随机数生成器的种子，否则对同一个神经网络重复拟合会产生不同的预测。

预测结果的变化程度取决于网络的复杂度，以及为避免过拟合而采取的措施，包括优化器的早停。在选择模型配置时，我们不仅要尽量降低预期错误率，还要尽量减小错误率的变异性。为此，我们力求降低测试错误率的 95% 预测区间上限。神经网络研究分为两个阶段：首先通过筛选实验找出最有用的（循环）神经网络，然后针对更多技能组，与常见预测方法开展更全面的性能研究。

第2节介绍了呼叫量的双随机模型，以及如何通过修改最初提出的计算方法来改善收敛性。第3节详细说明如何使用全因子设计，根据五个因素（及其交互作用）来评估不同循环神经网络方案对最终短期预测错误率的影响。此外，第3节还介绍了如何选择模型配置，使测试错误率的95%预测区间上界达到最小。鉴于需要运行的模型配置数量众多，且循环神经网络的计算复杂度较高，第3节讨论的第一阶段仅考虑数量有限的技能组和验证日。在第4节中，我们将在规模更大的验证集上，对所有呼叫中心技能组运行表现最佳的循环神经网络，并将其表现与双随机混合模型方法、自回归积分滑动平均模型以及温特斯平滑法进行比较。

2. 混合模型拟合的稳定设置

呼叫量受到两种不同因素的影响，这些因素会使观测到的呼叫次数之间产生相关性，从而违背普通最小二乘回归模型在对呼叫量建模时所采用的独立性假设（Ibrahim & L'Ecuyer，2013）。在某一天内，某些事件可能会导致呼叫量高于或低于预期。例如，上午股票市场出现意外波动，可能会导致某金融服务联络中心当天其余时段的呼叫次数增加。这属于日内相关性。同样，某些系统性过程也会导致日间相关性。举例来说，如果我们发现今天因天气事件等原因，残差全天都非常大且为正，那么我们可能也会预期明天的呼叫量高于平均水平。忽略连续观测值之间的相关性会导致标准误和预测区间不准确。此外，尽管存在相关残差时线性回归的估计值可能仍然无偏，但其效率会降低（Demidenko，2013）。

呼叫中心回归模型通常会纳入星期与时段交互作用（Ibrahim、Ye、L'Ecuyer 和 Shen，2016）。对于每周营业五天、每天有 32 个半小时时段的呼叫中心，这种交互作用涉及 160 个固定效应参数。此外，呼叫中心可能需要针对节假日进行预测。Aldor-Noiman、Feigin 和 Mandelbaum（2009）在训练模型时排除了节假日；然而，我们不能忽略这些日期，因为有些数据划分会在节假日运行，而且这些日期的行为可能有所不同。为捕捉这种行为，我们在模型中加入节假日指示变量（节假日_指示变量）与时段交互效应。

然而，有些训练数据集可能只包含一个节假日，导致该效应的参数估计方差较大（该日期的每个时段观测值都会成为节假日期间该时段的新估计值）。为降低这些估计值的变异性，我们在节假日期间将每 3 个时段合并为一组。也就是说，时段 {1, 2, 3} 被分配到 $p_$组 = 1，时段 {4, 5, 6} 被分配到 $p_$组 = 2，依此类推。该

$p_$组*节假日$^{-}$独立 交互项作为加性效应纳入固定效应结构。

沿用 Aldor-Noiman 等人（2009）的方法，我们对经过变换的呼叫次数拟合了一个带相关误差的线性混合模型

$$Y = \boldsymbol{X}\beta + \boldsymbol{Z}b + \varepsilon$$

其中

- Y 是经过变换的呼叫次数向量，$Y = \sqrt{count + 0.25}$
- $\boldsymbol{X}$ 是一个矩阵，其中包含每个观测值的固定效应水平
- β 是固定效应参数向量，其中包含星期几*时段交互作用和$p_$组*节假日指示变量交互作用
- $\boldsymbol{Z}$ 是模型中随机日间效应的二元系数矩阵。数据中的每个日期对应一列。
- $b \sim N(0, \boldsymbol{G})$ 是随机日间效应向量。数据集中的每个不同日期都由$b.$中的一个随机效应表示。$\boldsymbol{G}$遵循一阶自回归结构，即一阶自回归。
- $\varepsilon \sim N(0, \boldsymbol{R})$ 是误差项（残差）向量，允许 ε 在日期内可能遵循一阶自回归过程。因此，$\boldsymbol{R}$是一个分块对角矩阵，每个日期对应一个一阶自回归分块。这考虑了日期内相邻时段残差之间可能存在的相关性。

完整模型允许采用复杂的相关结构。然而，对于某些数据划分（在特定训练数据集中），呼叫到达可能仅零星且稀疏地出现。这在某些情况下会导致模型收敛缓慢或失败。Aldor-Noiman等人（2009）通过分两步估计双随机模型来解决这一问题：首先，使用每天汇总的呼叫次数估计日间相关性（$\boldsymbol{G}$）。随后，在第二次调用 SAS PROC MIXED 时，将这些参数固定，同时估计β 和$\boldsymbol{R}$ 。

事实上，当解位于参数空间的边界上时，例如方差分量为零时，PROC MIXED 可能会遇到收敛问题（Karl、Yang 和 Lohr，2013）。不过，在修改 PROC MIXED 的默认设置后，我们能够稳定地通过一次调用 PROC MIXED，对 $(\beta, \boldsymbol{G}, \boldsymbol{R})$ 进行联合优化，从而使完整模型收敛。在这方面，我们的方法与 Aldor-Noiman 等人（2009）的方法不同：我们对所有模型参数进行联合拟合（只需一次调用 PROC MIXED）。对于能够收敛的模型，这将降低估计值的偏差。

我们通过更改 SAS PROC MIXED 使用的收敛判据，提高了收敛率。默认情况下，SAS 会确保参数梯度平方之和（以当前参数估计的海森矩阵加权）足够小。然而，在双随机模型存在强相关性的情况下，参数估计可能接近参数空间的边界，这意味着随着收敛，梯度可能不会趋近于 0（Demidenko，2013）。作为替代方案，当迭代间对数似然的相对变化足够小时，我们便判定模型收敛。此外，我们在估计过程中采用费希尔评分法。对于具有复杂协方差结构的模型，费希尔评分法更加稳定，并能得到更好的渐近协方差估计（Demidenko，2013）。最后，由于我们的

应用只使用呼叫量的点估计，而不使用相关的标准误或显著性检验，因此我们指定自由度方法=残差，以避免耗费大量时间计算近似 *F* 检验所需的自由度。如果需要置信区间或预测区间，则应将此值设为 自由度方法=*Kenward-Roger*方法*2*，以计算自由度的 Satterthwaite 近似，并对标准误应用 Kenward-Rodger校正（Kenward & Roger，2009）。我们所用改进方法的代码见图1。

```
proc mixed data=training_data scoring=50 maxiter=150 maxfunc=10000 convf=1E-6;
    class day_of_week period day_num split p_group;
    by split;
    /* The fixed effects */
    model transf_call_count=day_of_week*period p_group*holiday_ind/
        noint ddfm=residual outp=pred_call_count_output notest;
    /* The day-level random effects */
    /* Note: day_num_copy is not included in the CLASS statment and is numeric */
    random day_num / type=sp(pow)(day_num_copy);
    /* The period-level correlated residuals */
    repeated period / type=ar(1) subject=day_num;
run;
```

图*1* 双重随机模型的修改版 *SAS* 代码

对观测到的呼叫量进行平方根变换，以减小其右偏程度，并稳定观测值的方差，因为呼叫量等数量通常服从泊松分布。图1中的方法对这一过程采用正态近似。我们尝试对未经变换的呼叫量拟合混合泊松回归（通过 PROC GLIMMIX），但发现运行时间长到难以接受（即使使用默认的伪似然方法并避免积分近似），而错误率并未明显改善。

3. 通过设计实验选择循环神经网络配置 t

一般而言，神经网络由权重层和非线性激活函数构成，用于关联输入（预测变量）与输出（目标值）。每一层的输出都会依次作为输入向量传递到下一层。每层的复杂程度由其生成的输出向量长度（节点数）决定。损失函数用于比较神经网络最后一层的输出与给定的目标值（例如呼叫量），优化器函数则会更新各节点的权重，以降低由此产生的损失。模型的“深度”由所用层数决定。“深度”一词正是“深度学习”这一说法的来源。例如，在采用卷积神经网络的图像处理应用中，可以发现不同层分别表示图像细节的不同粒度级别（Chollet & Allaire，2018）。除了层数和每层的节点数之外，还需要决定优化器的属性、参数权重随机初始化的分布，以及激活函数的形状。

在传统的密集连接网络中，通常假设各个观测值 e
相互独立。下面以 JMP Pro 14.1 的输出为例进行说明。假设我们要拟合一个密集连接神经网络，仅使用前一天同一时期的标准化呼叫次数（由于示例中每天有 32 个时期，因此即呼叫次数的滞后32期）作为预测变量，并在一个层中设置一个节点，使用双曲正切激活函数，以预测标准化呼叫次数。图2展示了该网络，图3展示了相应的权重。

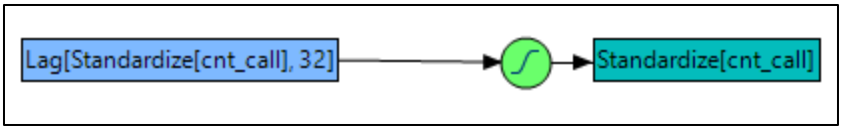


图2 单层单节点的密集连接神经网络。

Estimates

Parameter	Estimate
H1_1:Lag[Standardize[cnt_call], 32]	0.5323
H1_1:Intercept	0.4046
Standardize[cnt_call]_1:H1_1	3.3923
Standardize[cnt_call]_2:Intercept	-0.6354

图3 单层单节点网络的拟合权重。

假设在某个时期 t，滞后 32 期标准化呼叫次数等于 1。那么，神经网络预测的标准化呼叫量为

$$-0.6354 + 3.3923 * \mathrm{TanH}\big(0.5 * (0.4046 + 0.5323 * 1)\big) = 0.847$$

对于当前时期，t，其中 双曲正切函数 是双曲正切函数，0.5 参数为固定值。非线性激活函数使网络能够灵活地对输入与响应之间的非线性关系，以及输入之间的交互作用进行建模。

接下来，我们考虑一个稍复杂的网络，该网络有两层：第一层包含 2 个节点，第二层包含 1 个节点（图4），参数估计如图5所示。

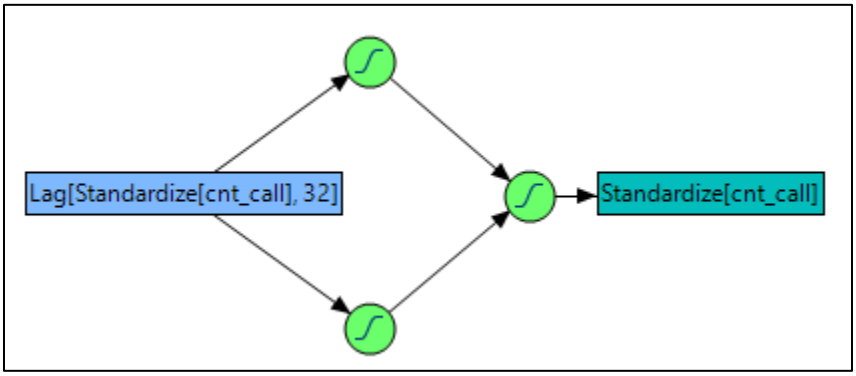


图4 一个具有两层的密集连接神经网络：第一层使用两个节点，第二层使用一个节点。

Estimates

Parameter	Estimate
H2_1:Lag[Standardize[cnt_call], 32]	-0.8785
H2_1:Intercept	-0.3633
H2_2:Lag[Standardize[cnt_call], 32]	0.4124
H2_2:Intercept	-0.0824
H1_1:H2_1	-0.7382
H1_1:H2_2	0.3057
H1_1:Intercept	0.2416
Standardize[cnt_call]_1:H1_1	4.9517
Standardize[cnt_call]_2:Intercept	-0.8200

图5 两层网络的拟合权重。

再次假设滞后 32 期标准化呼叫次数为 1，则预测值为$-0.82 + 4.9517$

$$* TanH\Big(0.5 * \Big(0.2416 - 0.7382 * TanH\big(0.5 * (-0.3633 - 0.8785 * 1)\big) + 0.3057 * TanH\big(0.5 * (-0.0824 + 0.4124 * 1)\big)\Big)\Big) = 0.843$$

在密集连接网络中，每个观测值都是独立处理的，不会“记住”先前处理的观测值中发生了什么。然而，在时间序列应用中，数据记录具有时间顺序，相邻观测值之间可能存在相关性。例如，呼叫量的激增或骤降可能会持续多个时期。为应对这种可能性，循环神经网络在拟合每个观测值时会记录信息，并将其作为模型输入，用于拟合后续观测值。

在简单（Elman）循环神经网络层中，每个节点的输出（即 双曲正切函数的输出，称为“状态”）都会被记录并存储，并在处理下一个观测值时作为同一节点的输入。请注意，该层的每个节点都有一个对应的状态记录。对于单层网络示例（图3），当滞后标准化呼叫量等于 1 时，状态计算结果为 s_t = 双曲正切函数$(0.5 * (0.4046 + 0.5323 * 1)) = 0.44$。简单循环神经网络会学习一个额外参数（例如，$u$），作为存储状态的系数；而密集网络使用的激活函数双曲正切函数$(0.5 * (w_0 + w_1 * X_t))$则会替换为 s_t = 双曲正切函数$(0.5 * (w_0 + w_1 * X_t + u * s_{t-1}))$，以拟合简单循环神经网络。长短期记忆网络和门控循环单元循环神经网络在对当前时间段进行预测时，也会使用记录下来的状态，同时结合其他激活函数的乘积，以便将状态信息传递到更远的未来。有关这些附加结构的详细信息，请参见 Bianchi 等人（2018）第3节。就本文的目的而言，只需指出门控循环单元网络与长短期记忆网络极为相似，只是由于省去了一组参数，复杂度较低。Chollet & Allaire（2018）指出，谷歌翻译目前使用具有七个大型层的长短期记忆网络运行。

事先无法确定这四种神经网络中哪一种最适合特定的呼叫中心。此外，将这些网络应用于呼叫中心数据时，每种网络可能都有不同的最优深度和结构。我们通过设计实验来确定最优模型类型和结构。

3.1 实验数据

我们分析了某运营呼叫中心中3个不同高呼叫量技能组在2018年3月至6月期间按30分钟时段汇总的呼叫量。所有待评估的模型均使用5周的训练数据来预测次日呼叫量。呼叫中心每天运营期间包含32个30分钟时段。本应用考察呼叫技能组周一至周五的行为。由于神经网络将滞后一周的观测值用作预测变量，因此第一周的训练数据不纳入预测变量矩阵（因为前一周的呼叫量未知），这意味着每个训练数据集包含4*5*32=640 个观测值。本节中的设计实验将使用留出期评估这些方法：留出期为6月最后一周内的5个单日提前预测，所用数据来自呼叫中心中呼叫量最大的三个技能组。随后，第4节将针对所有技能组拟合一组精简模型，并进行60天的提前预测。

3.2 神经网络输入因子

每个网络都会使用星期的独热编码（通过二元指示矩阵实现）以及一天中各时段的独热编码。此外，为捕捉日内和周内相关性，在对当前日期建模时，网络还会输入前一天同一时段的呼叫量，以及上周同一天同一时段的呼叫量。由于循环神经网络的序列内存正是用于此目的，因此不纳入滞后一个时段的呼叫量。其他输入包括：一个用于指示当前日期是否为节假日的二元指示变量、一个用于指示昨天是否为节假日的二元指示变量、一个用于指示上周的观测值是否记录在节假日的二元指示变量，以及日序号。日序号是数据集中某一日期编号的连续计数器，神经网络或许可以借此检测随时间变化的趋势。总计而言，这些输入构成了神经网络的 42 个输入向量。无需为星期与时段之间的交互作用（或与其他因素的交互作用）创建指示变量列，因为神经网络会自动检测这些交互作用。

Bianchi等人（2017年）对每小时呼叫量的应用显示出显著的滞后24期相关性，体现了时段效应。他们通过对滞后24期的数据进行差分来消除这种季节性。相比之下，我们不对呼叫量进行差分，而是将时段（以及星期几）作为外生变量，并让神经网络识别这种季节性。这种方法使网络能够识别时段与星期几之间预期的交互作用，以及其他任何输入因子。

虽然所有模型都包含这些输入向量，但还有最后三个输入向量，其（联合）纳入与否被视为一个实验因子：来自混合模型方法的同期预测（Aldor-Noiman, Feigin, & Mandelbaum，《呼叫中心工作量预测：方法论与案例研究》，2009）、来自 Winters 平滑模型的预测，以及来自季节性自回归积分滑动平均模型 ARIMA(1, 0, 1)(0, 1, 1)160 的预测。将这些模型的预测作为神经网络的输入，是一种原创方法，它使网络有机会根据与其他纳入因素之间可能存在的交互作用，形成可视为对这些传统模型预测进行修正的预测。为简洁起见，我们将此称为 mixed.cheat 选项，因为它允许神经网络在生成自身对相同时期的预测时，参考其他这三个模型生成的预测，从而“作弊”。如果不将其他 42 个输入向量中的任何一个与这三个输入向量一同纳入，那么这就相当于采用监督学习方法，对这三个模型的预测进行动态加权平均，从而生成一个单一的“集成”预测。

3.3 作为实验因素处理的网络配置方面

该设计实验考察了神经网络结构设置的五个不同实验因子：模型类型 {密集型、简单（Elman）循环神经网络、门控循环单元、长短期记忆网络}，层数 {1, 2 25, 50, 75, 100*20*, 00001 }，节点数 （每层） { }，核*.L .reg* { . } 以及 *mixed.cheat* {FALSE, TRUE}。L2 正则化项会将每个权重系数乘以核 *L2* 正则化的结果加到网络的总损失函数中。与套索回归类似，这有助于限制模型系数的大小，并有可能防止训练数据过拟合。

我们针对这五个因素拟合了一个全因子设计（需要运行 128 次），这使我们能够检验这五个因素之间是否存在最高五阶的交互作用。图 6 展示了该设计的前 6 次运行。我们在最大的 3 个数据划分上重复该设计，并针对随后 5 个提前一天的预测进行重复。这使整个实验的运行总数达到 1920 次。重复实验可以对不同日期的表现取平均值，并更详细地探究错误率的方差如何取决于各个因素。

model.type	nlayers	mixed.cheat	nnodes	kernel.L2.reg
layer_simple_rnn	2	FALSE	50	0.0001
layer_gru	1	FALSE	25	0
layer_lstm	1	TRUE	100	0
layer_gru	1	TRUE	75	0.0001
layer_gru	1	TRUE	75	0
layer_simple_rnn	2	FALSE	100	0

图 *6* 设计实验的前 *6* 次运行

3.4 神经网络配置的静态考量

经典全连接神经网络的输入是一个 640x42 矩阵（如果 混合作弊=*TRUE)*，则为 640x45）。全连接网络不考虑 640 个观测值的时间顺序（除显式将日期和周滞后观测值作为输入外）：每个训练轮次后都会打乱观测值，然后分批处理（我们使用的批量大小为 32）。

相比之下，循环神经网络会考虑观测值的顺序。循环神经网络的输入是一个 20x32x42 数组。这表示有 20 个批次（日期），每个批次包含 32 个时间步（时期），每个时间步有 42 个预测变量。默认情况下，各批次彼此独立处理，而每个批次内的时间步可能彼此相关（通过循环神经网络中状态的持续性）。如果批次本身按时间顺序输入（如我们的应用场景），则可通过 Keras 模型的 STATEFUL=TRUE 选项，并在训练期间禁用批次打乱来指定。这样可以在批次之间（日期之间）保留模型权重，以捕捉可能存在的长期行为。不过，我们发现，使用有状态选项会导致某些技能*日期组合无法收敛，而且得到的验证误差率与不使用有状态选项时相比并无显著差异。这似乎表明，通过纳入滞后一天和滞后一周的观测值，已经捕捉到了对前几日行为的依赖。由于偶尔会出现收敛问题，后续结果均使用 STATEFUL=FALSE 生成（训练期间启用批次打乱）。也可以通过在 4x160x42 或 1x640x42 数组上训练循环神经网络，使用周长或月长批次拟合模型。我们没有考虑周长批次，但

月长批次得到的预测往往不如日长批次。这可能是因为数据中缺乏长期相关性，而且较小的批次允许打乱日期输入神经网络梯度优化程序的顺序；这可以防止模型对每个训练轮次中最先输入网络的观测值赋予过高权重，从而改善模型拟合效果（Chollet & Allaire，2018）。

请注意，对于每个技能组，都会重置模型权重并重新训练模型。这与 Zhu & Laptev（2017）所采用的方法形成对比：他们拟合单个网络，以适应不同城市之间迥异的行为。正如第5节所讨论的，理论上可以构建一个单一的多输出网络，同时对所有技能组进行建模。

虽然在本应用中没有将它们纳入实验因素，我们也注意到预测质量与所采用的优化程序之间存在显著关系。经过大量的初步实验，我们采用了 Adam优化器（Kingma & Ba，2014）的 AMSGrad 变体（Reddi, Kale, & Kumar，2018），并设置学习率衰减为 0.0001。我们建议在未来问题的参数调优设计实验中，将优化器和可选的学习率衰减都纳入实验因素。

我们发现，使用验证数据（训练集中的最后一周）为每次模型拟合调整训练轮次数至关重要。具体做法是：先为每个模型拟合 500 个训练轮次，对验证数据上的加权绝对百分比误差取移动平均（窗口大小为 10 个训练轮次），然后根据使验证加权绝对百分比误差达到最小值的训练轮次数重新拟合模型（同时使用训练数据和验证数据，以预测额外留出作为测试集的一个日期）。由于我们观察到单次记录的加权绝对百分比误差波动较大且并非单调变化，移动平均十分重要，有助于找到相对稳定的区间。与先前的研究结果一致（Bianchi, Maiorino, Kampffmeyer, Rizzi, & Jenssen，2017），我们注意到，循环神经网络收敛所需的训练轮次数远多于稠密神经网络。其他研究人员（例如 Bianchi 等人）在拟合循环神经网络时使用了超过 500 个训练轮次，因此在构建循环神经网络时，也应将这一上界视为重要因素。

我们也尝试使用循环 dropout（Gal & Ghahramani，2015）来防止过拟合，但在实验初期，这种方法导致性能下降，因此我们不再考虑采用它。不过，这或许只是由该特定应用的某些特点造成的，例如训练期相对较短，仅为五周：Chollet 和 Allaire（2018）强烈建议使用 dropout 和循环 dropout。

将随机权重的核初始化器改为 He正态初始化器（He, Zhang, Ren, & Sun，2015）后，模型的错误率有所下降。我们始终只使用 ReLU激活函数，不过这一选择也可能影响最终模型拟合的质量。尽管我们的应用没有检测到五个实验因素之间存在任何双因素交互作用，但某些额外因素也可能取决于所用的（循环）神经网络类型，这意味着这些因素与模型类型之间存在交互作用。

最后一个影响因素是批量大小。它决定了在更新梯度之前会处理多少个观测值：使用图形处理器训练 Keras 模型时，图形处理器的可用内存量

可能会限制批量大小。对于循环神经网络，这一选择受到更多限制，因为每个批次都是一个日期/周/月（由循环神经网络输入数组中指定的时间步数量决定）。相比之下，全连接网络的批量大小可以设为 1 到训练数据中观测值数量之间的任意值。我们注意到，全连接模型的错误率会因所用批量大小不同而显著变化。

3.5 实验结果

虽然均方误差（MSE）常用于评估和比较预测模型，但对于呼叫中心应用而言，它并不是一个理想的指标，因为它会过度关注每天开始和结束时呼叫量较低的时期。因此，建议在呼叫量建模中使用加权绝对百分比误差（WAPE）（Ibrahim, Ye, L'Ecuyer, & Shen, 2016）。该指标以每个时期收到的呼叫数量为权重，对该时期的绝对百分比误差进行加权，定义如下：

$$WAPE = \frac{\sum_{i=1}^{n} |Y_i - \hat{Y}_i|}{\sum_{i=1}^{n} Y_i}$$

其中，Y_i 和 $\hat{Y}_i$ 分别表示每个 30 分钟时期 $i = 1, \ldots, n$ 的观测呼叫量和预测呼叫量。由于 WAPE 是我们关注的指标，因此编译模型时采用平均绝对误差损失函数，以最小化 WAPE 的分子（分母为固定值）。

对于实验表（图 6）中的每一行，都会将指定的神经网络分别拟合到每个数据划分中连续五天的数据。也就是说，使用截至第 1 天之前的 5 周数据来预测第 1 天，然后重置模型，再使用截至第 2 天之前的 5 周数据来预测第 2 天，依此类推。将五个单日提前预测组成的向量与观测到的呼叫量（训练期间模型无法看到这些数据）进行比较，并记录所得的加权绝对百分比误差。

3.6 实验结果分析

图 7 展示了不同神经网络配置下预测误差的典型表现。门控循环单元通常具有最低的预测误差，或者至少始终接近最低值。其他方法的表现往往不够稳定，这取决于节点数和层数的选择，也会因日期和数据划分而异。

		model.type			
		NN Classic	RNN_GRU	RNN_LSTM	RNN_Simple
		WAPE	WAPE	WAPE	WAPE
nlayers	nnodes	Mean	Mean	Mean	Mean
1	25	6.7%	6.2%	8.5%	6.0%
	50	7.1%	6.2%	6.5%	6.7%
	75	7.1%	5.8%	6.8%	5.9%
	100	7.0%	6.4%	6.9%	6.8%
2	25	7.5%	6.6%	8.1%	6.9%
	50	7.3%	5.8%	6.9%	7.0%
	75	7.1%	6.0%	6.6%	6.5%
	100	8.0%	5.9%	7.5%	6.0%

图 7 预测日期为 5、数据划分为 3 时，不同模型类型及设置下的预测误差

回归分析用于确定具有统计显著性的因素和交互作用。为了控制记录的加权绝对百分比误差严重右偏的情况，采用了倒数

变换，并将其用作回归模型中的响应。图12展示了原始数据的偏斜情况，以及对加权绝对百分比误差取倒数后经变换达到正态分布的结果。

3.7 对数线性方差回归模型

通过纳入五个实验因素（模型类型、层数、混合作弊、节点数、核 *L2* 正则化）的主效应及所有交互作用，对 1/WAPE 的均值结构进行建模。此外，还将 数据划分、文件 （代表验证日期）以及 数据划分*文件作为区组因素纳入模型。

全因子设计的普通回归模型会假设设计空间中所有响应的误差方差相同，而对数线性方差模型则允许误差方差本身作为输入因子的函数进行建模。对于本应用而言，这一做法很有吸引力，因为预计简约网络在重复拟合期间表现出的变异性可能较小。

对于每种因素设置，对数线性方差模型都会生成 1/WAPE 的预测均值和预测标准差，表示神经网络随机初始权重所导致的变异性。

我们将 模型类型、层数、混合作弊、节点数、核 *L2* 正则化、文件、和 数据划分作为因素纳入对数线性方差模型（使用 JMP Pro 14.1）。该模型将误差方差的对数建模为这些因素的线性组合（所有因素均视为分类变量）。除 核 *L2* 正则化外，所有这些效应均被发现与误差方差显著相关（图8）。

Variance Effect Likelihood Ratio Tests

Source	Test Type	DF	ChiSquare	Prob>ChiSq
model.type	Likelihood	3	114.126	<.0001*
nlayers	Likelihood	1	4.6333	0.0314*
mixed.cheat	Likelihood	1	53.6884	<.0001*
nnodes	Likelihood	3	16.5624	0.0009*
kernel.L2.reg	Likelihood	1	0.0042	0.9482
file	Likelihood	4	36.8213	<.0001*
split	Likelihood	2	56.3123	<.0001*

图*8* 对 *WAPE*变异性有显著影响的因素

对于均值模型，模型类型、层数、混合作弊、节点数、日期、数据划分和文件*数据划分均被发现与 1/WAPE 显著相关（图9）。核 *L2* 正则化以及涉及神经网络架构的任何交互项均未达到显著水平。

Fixed Effect Tests

Source	Nparm	DF	DFDen	F Ratio	Prob > F
model.type	3	3	796.5	44.6120	<.0001*
nlayers	1	1	1258	5.5213	0.0189*
model.type*nlayers	3	3	796.5	1.0691	0.3614
mixed.cheat	1	1	1258	17.0652	<.0001*
model.type*mixed.cheat	3	3	796.5	1.3181	0.2672
nlayers*mixed.cheat	1	1	1258	2.3413	0.1262
model.type*nlayers*mixed.cheat	3	3	796.5	1.9528	0.1196
nnodes	3	3	786.2	4.5600	0.0036*
model.type*nnodes	9	9	839.4	0.7864	0.6290
nlayers*nnodes	3	3	786.2	0.7371	0.5301
model.type*nlayers*nnodes	9	9	839.4	0.6668	0.7395
mixed.cheat*nnodes	3	3	786.2	0.6624	0.5753
model.type*mixed.cheat*nnodes	9	9	839.4	0.2224	0.9913
nlayers*mixed.cheat*nnodes	3	3	786.2	0.9291	0.4261
model.type*nlayers*mixed.cheat*nnodes	9	9	839.4	0.4312	0.9186
kernel.L2.reg	1	1	1258	0.0026	0.9590
model.type*kernel.L2.reg	3	3	796.5	0.1263	0.9445
nlayers*kernel.L2.reg	1	1	1258	0.2313	0.6306
model.type*nlayers*kernel.L2.reg	3	3	796.5	0.7812	0.5046
mixed.cheat*kernel.L2.reg	1	1	1258	0.0295	0.8636
model.type*mixed.cheat*kernel.L2.reg	3	3	796.5	0.1395	0.9364
nlayers*mixed.cheat*kernel.L2.reg	1	1	1258	0.1498	0.6988
model.type*nlayers*mixed.cheat*kernel.L2.reg	3	3	796.5	0.3776	0.7692
nnodes*kernel.L2.reg	3	3	786.2	0.9118	0.4347
model.type*nnodes*kernel.L2.reg	9	9	839.4	0.8905	0.5331
nlayers*nnodes*kernel.L2.reg	3	3	786.2	0.7442	0.5259
model.type*nlayers*nnodes*kernel.L2.reg	9	9	839.4	0.6535	0.7513
mixed.cheat*nnodes*kernel.L2.reg	3	3	786.2	0.0812	0.9703
model.type*mixed.cheat*nnodes*kernel.L2.reg	9	9	839.4	0.2691	0.9827
nlayers*mixed.cheat*nnodes*kernel.L2.reg	3	3	786.2	0.8510	0.4662
model.type*nlayers*mixed.cheat*nnodes*kernel.L2.reg	9	9	839.4	0.6614	0.7442
file	4	4	668.1	1258.316	<.0001*
split	2	2	857.3	1867.911	<.0001*
file*split	8	8	746.8	244.5502	<.0001*

图*9* 与平均加权绝对百分比误差显著相关的因素

移除不显著的交互项（但允许核 *L2* 正则化继续保留在均值模型和方差模型中）后，得到图 10 所示的剖析器。请注意，LSTM 模型对应的标准差很大。由于模型中没有将文件和数据划分与其他实验因素之间的交互作用纳入建模，因此未显示它们的依赖关系。我们希望**最大化 1/WAPE**，同时也希望**最小化 1/WAPE 的标准差**。也就是说，为了找到一种模型配置，使其平均结果相对较好，同时能够避免较大的误差，我们希望最大化 1/WAPE 的 95% 下预测区间（或最小化 WAPE 的 95% 上预测区间）。

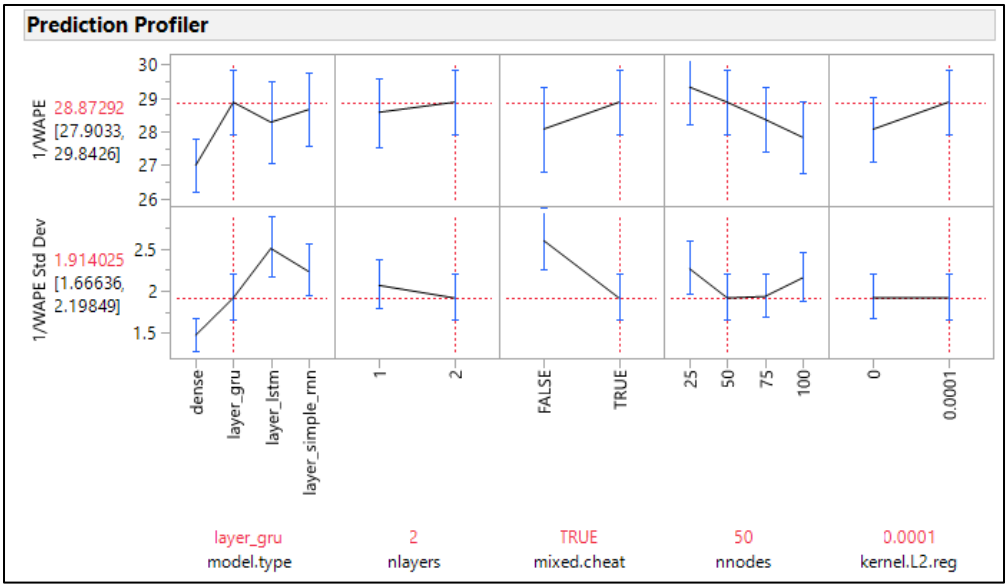


图*11 1/WAPE* 的均值和标准差随实验因素的变化。目标是最大化上方的数值（*1*/平均预测误差），并最小化下方的数值（标准差）

图11绘制了 WAPE 的上侧 95% 预测区间与实验因素的关系。使 WAPE 上侧 95% 预测区间最小的模型配置是一个 GRU 模型，它可以使用混合模型、自回归积分滑动平均模型和 Winters 预测作为输入，包含 2 层，每层有 50 个节点，并对核权重施加 0.0001 的 L2惩罚。然而，无论是在均值模型还是方差模型中，均未发现 L2惩罚具有显著性，并且 层数 的影响相对平缓。

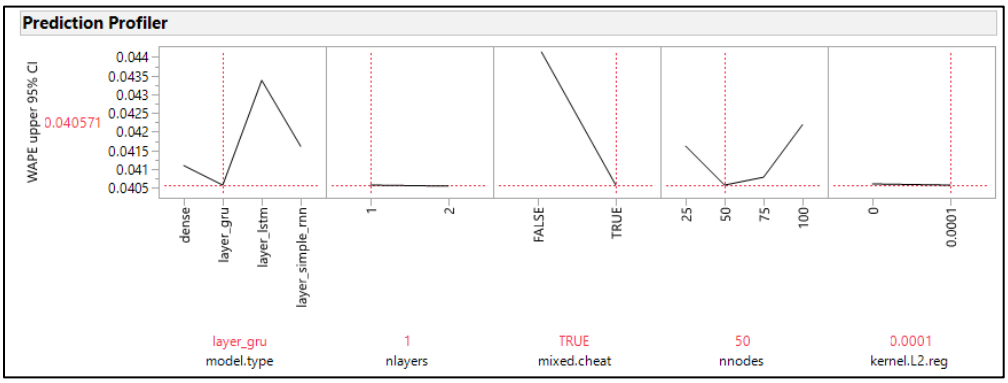
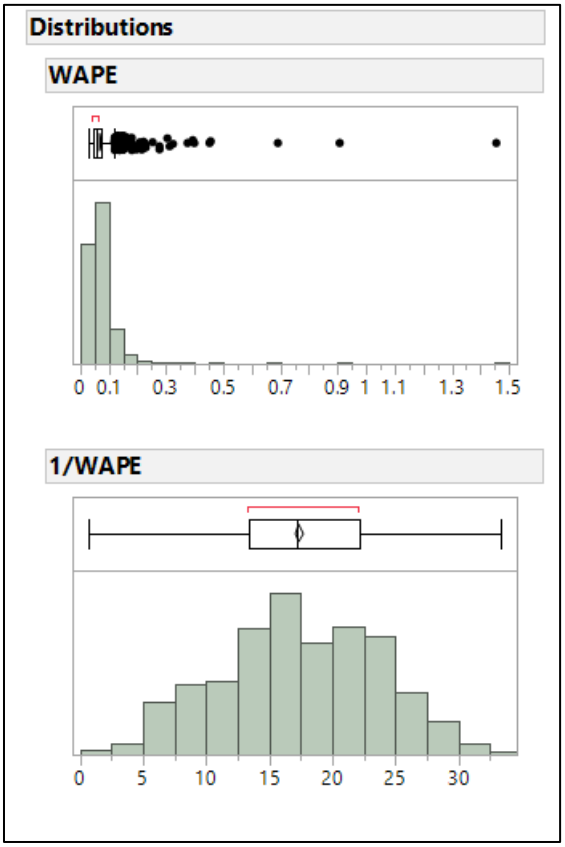


图*11 WAPE* 的上侧 *95%* 预测区间

总而言之，设计实验为如何在多个神经网络模型和参数中有效选择适当的水平提供了参考。目标是在尽量减小预测误差（即最大化 1/预测误差）和尽量减小该预测误差的方差之间取得平衡。由于因素之间不存在显著交互作用，因此图 10 和图11所示的所有因素和设置均可进行解读。图 10 上半部分显示倒数预测误差（值越大越好），表明模型首选门控循环单元或埃尔曼（简单）、1 层、采用混合模型预测和 25 个节点，而选择 L2.kernel 几乎没有影响。图 10 下半部分显示方差（越低越好），结果表明，传统神经网络更为理想，且应采用 2

层、混合模型预测和 50 个节点，同时对正则化参数的取值具有稳健性。请注意，传统稠密神经网络在所有情境下的预测误差始终较差，因此尽管其方差最低，也不建议采用。预测误差的预测区间是许多从业者更偏好的另一种视角，其目标是尽量缩小区间宽度。图11（越低越好）清楚表明，采用混合模型预测和 50 个节点时，门控循环单元是首选方案。



设计实验得出的加权绝对百分比误差及其倒数的代表性分布

4. 全面性能研究

基于神经网络误差率设计实验的结果，我们使用覆盖36个技能组而非仅3个技能组的实际呼叫中心数据，开展了进一步研究。我们采用一致的5周训练期，在数月的数据中逐步推进，以生成60个单日提前预测。这些预测在模型训练期间未使用预测日期的实际数据，因此可视为训练后模型的验证集。这使我们能够比较混合模型与神经网络的相对性能。我们还纳入了自回归积分滑动平均模型和温特斯季节性平滑模型的错误率，以供比较。

尽管根据设计实验，我们本可以简单地只选择每层有 50 个节点、共 2 层且核 *L2* 正则化 复杂度参数 设为 0.0001 的 GRU 循环神经网络作为循环神经网络的代表，但我们决定纳入全部 4 种神经网络方法

这些方法均使用相同的参数设置进行筛选。由于新增了技能组——其中许多技能组的呼叫量要低得多——另一种循环神经网络模型可能会比 GRU 表现更好。

4.1 结果：60 个独立验证日的单日提前预测

图13展示了呼叫量最高的12个数据划分在全部60天中的平均预测误差，并按呼叫量从高到低排序。请注意，本节结果由未启用混合作弊选项的（循环）神经网络生成。除数据划分5540外，双随机混合模型的平均预测误差均为最低，而且通常显著低于其他方法。考虑到其相对简单的特点，温特斯季节性指数平滑的表现也相当出色。门控循环单元的表现证实了设计实验的结果：它通常是表现最好的循环神经网络，而且始终具有竞争力。复杂度极高的长短期记忆网络在其中许多数据划分上的预测误差都非常大，因此不建议使用。另请注意，随着呼叫量下降，所有方法的预测误差总体上都会增加。

Split	Sum Call Vol	ARIMA	Doubly Stoch	NN_Classic	RNN_GRU	RNN_LSTM	RNN_Simple	Winters
5000&5240	929754	9.0%	8.4%	10.2%	11.2%	38.0%	13.9%	8.3%
5400&5570	461256	7.4%	6.6%	8.8%	8.4%	26.1%	14.3%	7.1%
5620	162996	8.6%	7.7%	9.6%	9.1%	28.7%	9.6%	8.4%
5660	137759	10.2%	10.1%	13.2%	12.0%	17.1%	14.0%	10.5%
5020	71930	11.5%	11.2%	14.5%	12.3%	27.3%	14.2%	11.7%
5630	65079	15.4%	14.9%	16.7%	15.5%	26.4%	15.5%	14.1%
5840	63984	13.1%	12.1%	14.6%	13.4%	25.3%	12.9%	13.0%
5200	48728	13.2%	13.4%	15.6%	13.5%	52.4%	13.5%	13.6%
5540	38236	15.9%	15.8%	16.8%	16.3%	28.5%	16.6%	16.0%
5670	34793	14.0%	13.2%	15.6%	14.2%	27.3%	13.9%	14.0%
6500	30534	16.8%	16.4%	18.5%	17.3%	23.0%	18.0%	17.1%
5260	23849	21.9%	20.9%	22.7%	20.3%	32.5%	20.2%	21.2%

图*13* 高呼叫量数据划分在*60*个独立验证日上的平均加权绝对百分比误差

*c*全量数据划分

图14显示，在中等呼叫量的数据划分中，错误率也呈现出随着呼叫量下降而上升的类似趋势。门控循环单元通常优于其他神经网络方法，其错误率也更接近双随机混合模型。

Split	Sum Call Vol	ARIMA	Doubly Stoch	NN_Classic	RNN_GRU	RNN_LSTM	RNN_Simple	Winters
5460	20922	18.2%	18.1%	19.5%	18.1%	26.0%	17.7%	18.2%
5410	19461	19.8%	19.2%	21.4%	20.0%	59.1%	20.8%	19.7%
6350	16874	30.8%	26.8%	29.3%	28.7%	31.1%	29.0%	29.4%
5060	16765	19.3%	19.1%	20.7%	19.6%	24.4%	19.6%	19.6%
5650	9911	23.8%	23.4%	24.6%	24.0%	27.3%	24.3%	24.0%
5030	8102	40.0%	26.0%	28.5%	27.8%	33.2%	27.7%	26.9%
5680	7525	27.3%	27.1%	28.2%	26.7%	31.9%	27.8%	27.7%
5440	7402	28.5%	28.1%	30.3%	28.3%	33.8%	29.7%	29.2%
5070	6446	34.6%	34.5%	34.6%	33.5%	43.6%	35.0%	34.7%
5420	5247	35.0%	34.6%	36.2%	33.9%	38.1%	34.7%	35.0%
5899	4844	36.4%	35.8%	37.0%	35.2%	64.8%	36.6%	36.7%
5100	4019	34.7%	34.0%	36.6%	34.3%	40.1%	35.5%	34.8%

图*14* 中等呼叫量数据划分在*60*个独立验证日期上的平均加权绝对百分比误差

对于图15所示的低呼叫量数据划分，门控循环单元和简单循环神经网络的表现相近，且在大多数数据划分中略优于双重随机和 Winters。极低呼叫量（最后3个数据划分）似乎确实能从循环神经网络的形式中受益。

Split	Sum Call Vol	ARIMA	Doubly Stoch	NN_Classic	RNN_GRU	RNN_LSTM	RNN_Simple	Winters
5710	3742	39.6%	38.8%	40.7%	39.1%	45.7%	40.2%	38.9%
5820	2949	44.1%	43.1%	46.1%	43.2%	50.4%	44.0%	43.4%
5690	2238	56.1%	51.1%	51.6%	51.2%	58.9%	51.8%	52.5%
5220	2089	49.7%	49.5%	50.5%	49.3%	54.0%	50.1%	49.7%
5470	1975	49.2%	48.8%	52.0%	50.5%	57.0%	50.7%	50.2%
6330	1556	60.5%	60.1%	61.0%	60.2%	65.1%	61.6%	60.7%
5040	1398	79.0%	69.0%	69.8%	73.4%	72.2%	71.1%	80.7%
6310	922	98.5%	92.0%	92.2%	88.1%	91.0%	89.5%	92.8%
5720	918	80.2%	77.8%	77.5%	77.6%	85.0%	78.4%	80.1%
6370	485	95.3%	93.6%	96.5%	91.3%	111.3%	94.0%	95.2%
6360	171	150.4%	136.8%	126.5%	107.7%	118.7%	111.7%	142.5%
6340	14	189.7%	161.4%	188.2%	102.7%	108.9%	107.8%	187.7%

图*15* 低呼叫量数据划分在*60*个独立验证日上的平均加权绝对百分比误差预测误差

总体而言，表现最佳的方法是双随机混合模型和 GRU循环神经网络。图16按呼叫量对数据划分排序，展示了各划分的预测误差。总体来看，混合模型在高、中呼叫量的数据划分中表现更好（图中数值越低越好），而循环神经网络 model 对小呼叫量更为有效，尤其是对呼叫量极小的数据划分。

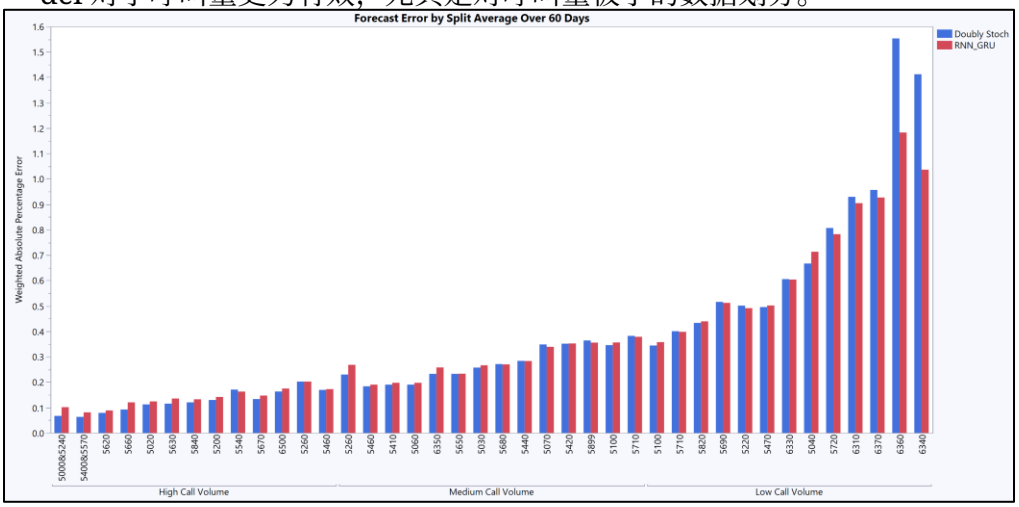


图*16* 按数据划分中的呼叫量排序的双重随机（蓝色）和门控循环单元（红色）的预测误差率

图17从另一个角度展示了这一相同模式。对于每个数据划分中的60个提前一天的预测，记录 GRU 和双重随机模型的加权绝对百分比误差，以及该数据划分在训练数据和验证数据中记录的呼叫总数（总_呼_叫）。对于每个数据划分，图17绘制了60个提前一天的预测中 GRU“胜出”（GRU 加权绝对百分比误差 < 双重随机模型加权绝对百分比误差）的百分比，并将其与该数据划分在60组不同训练数据和验证数据中的呼叫量中位数的对数进行对比。相对于双重随机模型，GRU 的表现似乎会随着呼叫量对数的增加而线性下降。

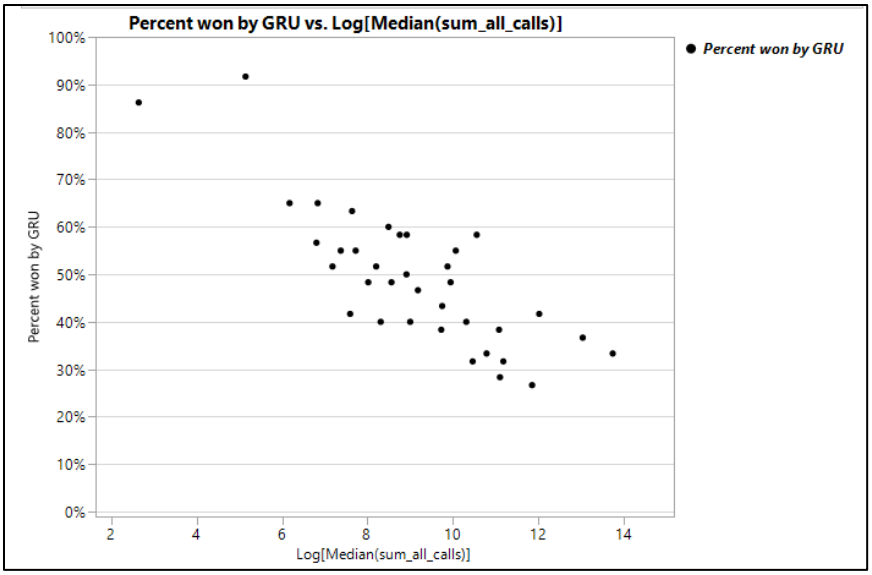


图*17*：在*36*个数据划分中，每个数据划分的*60*个提前日期预测里，*GRU* 加权绝对百分比误差低于双重随机加权绝对百分比误差的预测所占百分比；图中以每个训练数据与验证数据组合中记录的所有呼叫次数总和的中位数的对数为横轴。

对于呼叫量较大的数据划分，GRU 模型的额外灵活性并未带来优于混合模型方法所生成预测的效果。由于混合模型的计算速度更快，且实现起来更简单，因此在这一短期应用中，对于高呼叫量的数据划分，运行神经网络似乎并无益处。这与 Uber 交通流量研究在考察短期预测时的发现一致（Zhu & Laptev，2017年）；如果呼叫中心数据采用更长的训练期（数月或数年），结果可能会有所不同。

4.2 将双重随机预测作为协变量来改进 GRU循环神经网络

根据设计实验期间的初步研究，将双重随机模型、自回归积分滑动平均模型和Winters模型对验证日的预测值纳入其中，通常可以提升GRU循环神经网络的预测性能。这种利用其他模型预测值的“作弊”（如混合作弊)所示），对于GRU 模型在每种技能上的这60次预测而言，已证明具有显著优势。留出验证数据的加权绝对百分比误差仍是主要的性能衡量指标。

图18展示了各神经网络使用混合预测时，高呼叫量技能组在60个验证日上的平均预测误差。请注意并排比较 RNN_GRU（不作弊）列和 GRU¯作弊列：大多数情况下，“作弊”确实能改善预测；而在预测效果没有提升的情况下，下降幅度也很小。此外，Simple¯作弊的错误率与 GRU_作弊相比表现不相上下，而 LSTM¯作弊的性能则明显较差，且存在稳定性问题。双重随机预测仍然表现相当出色，而且通常是最佳选择。在考察中、低呼叫量的数据划分时，结果也一致。因此，我们建议将双重随机模型或温特斯模型的预测作为循环神经网络的输入。

Split	Sum Call Vol	Doubly Stoch	RNN_GRU	GRU_cheat	LSTM_cheat	Simple_cheat
5000&5240	926493	6.7%	10.2%	7.8%	14.4%	7.8%
5400&5570	459961	6.3%	8.1%	7.4%	9.4%	6.9%
5620	164138	7.9%	8.9%	8.8%	15.2%	8.7%
5660	138389	9.2%	12.0%	9.9%	18.1%	9.8%
5020	71900	11.2%	12.4%	12.4%	27.0%	12.3%
5630	65279	11.5%	13.5%	13.7%	421.5%	13.2%
5840	63687	12.0%	13.2%	13.1%	26.1%	12.7%
5200	48624	13.0%	14.2%	14.4%	28.1%	14.0%
5540	38318	17.1%	16.3%	16.9%	20.8%	17.5%
5670	34741	13.4%	14.7%	14.4%	24.5%	14.2%
6500	30352	16.3%	17.5%	16.8%	35.2%	17.2%
5260	23674	20.7%	21.3%	20.5%	28.0%	20.8%

图*18* 允许神经网络“作弊”以改进预测时，高呼叫量数据划分在*60*个验证预测日上的平均预测误差

GRU 加权绝对百分比误差（混合作弊=FALSE）减去 GRU 加权绝对百分比误差（混合作弊=TRUE）的中位数（针对每个数据划分/日期组合，样本量为 36*60=2160）为 0.002。Wilcoxon 符号秩检验得出的 p 值为 0.0005；该检验的零假设是这些差值来自中位数等于 0 的总体。这表明，混合作弊确实倾向于提升 GRU 模型的性能。

对错误率配对差值进行的类似比较（均设 mixed.cheat=FALSE）证实，门控循环单元优于其他（循环）神经网络：长短期记忆网络 - 门控循环单元的中位数为 0.0247，p值为1e-129；经典神经网络 - 门控循环单元的中位数为0.0690，p值为 <1e-185；简单循环神经网络 - 门控循环单元的中位数为0.0021，p值为1e-04。

参考文献

Aldor-Noiman, S., Feigin, P. D., & Mandelbaum, A.（2009）。呼叫中心工作量预测：方法论与案例研究。《应用统计年鉴》，*3(4)*，1403-1447。

Aldor-Noiman, S., Feigin, P. D., & Mandelbaum, A.（2009）。呼叫中心工作量预测：方法论与案例研究。《应用统计年鉴》(4)，1403– 1447。

Allaire, J., & Chollet, F. (2018). *keras*：*R* 接口，适用于*“Keras”*。取自 CRAN：https://CRAN.R-project.org/package=keras Bianchi, F. M., Maiorino, E., Kampffmeyer, M. C., Rizzi, A. R., & Jenssen, R. (2017). 用于短期负荷预测的循环神经网络：概述与比较分析。 Cham：Springer。Bianchi, F. M., Maiorino, E., Kampffmeyer, M. C., Rizzi, A. R., & Jenssen, R. (2018). 用于短期负荷预测的循环神经网络：概述与比较分析。*arXiv*。取自 https://arxiv.org/abs/1705.04378 Box, G., & Jenkins, G. (1970). 时间序列分析：预测与控制。 旧金山，加利福尼亚州：Holden-Day。

Cho, K., Merrienboer, B. V., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). 使用 RNN 编码器-解码器进行统计机器翻译的短语表示学习。*arXive*，arXiv:1406.1078。Chollet, F., & Allaire, J. (2018). *R* 深度学习。 谢尔特岛：Manning Publications Co. Demidenko, E. (2013). 混合模型理论及其在 *R* 中的应用。 新泽西州霍博肯：John Wiley & Sons, Inc. El man, J. L. (1990).《在时间中寻找结构》。《认知科学》，*14*，179-211。Gal, Y., & Ghahramani, Z. (2015). 循环神经网络中 Dropout 的理论基础应用。*ar Xiv*。取自 https://arxiv.org/abs/1512.05287 Gans, N., Koole, G., & Mandelbaum, A. (2003). 电话呼叫中心：教程、综述与研究展望。《制造与服务运营管理》，5:79-141。Gans, N., Koole, G., & Mandelbaum, A. (2003). 电话呼叫中心：教程、综述与研究展望。《制造与服务运营管理》，79-141。Harvey, A. (1990). 预测、结构时间序列模型与卡尔曼滤波器。 伦敦：剑桥大学出版社。He, K., Zhang, X., Ren, S., & Sun, J. (2015). 深入研究整流器：在 ImageNet 分类上超越人类水平的表现。*arXiv*。取自 https://arxiv.org/abs/1502.01852 Hochrieter, S., & Schmidhuber, J. (1997). 长短期记忆。《神经计算》，*9(8)*，1735-1780。Ibrahim, R., & L'Ecuyer, P. (2013). 呼叫中心到达量预测：固定效应、混合

-

效应和双变量模型。《制造与服务运营管理》 *t*,

72-85。Ibrahim, R., Ye, H., L'Ecuyer, P., & Shen, H. (2016). 呼叫中心到达量的建模与预测：文献综述与案例研究。《国际预测学杂志》, 865-874。Karl, A. T., Yang, Y., & Lohr, S. L. (2013). 多重隶属线性混合模型的高效最大似然估计及其在教育增值评估中的应用。《计算统计与数据分析》, *59*, 13-27。Kenward, M. G., & Roger, J. H. (2009). 精度的改进近似

f

基于限制最大似然的固定效应。计算统计与数据分析(53), 2583–2595。Kingma, D., & Ba, J. (2014). Adam：一种随机优化方法。*arXiv*。获取自 https://arxiv.org/abs/1412.6980v8

Laptev, N., Yosinski, J., Li, L. E., & Smyl, S. (2017). Uber 基于神经网络的时间序列极端事件预测。*ICML 2017* 时间序列研讨会。悉尼。获取自 http://roseyu.com/time-series- workshop/submissions/TSW2017_paper_3.pdf Reddi, S., Kale, S., & Kumar, S. (2018). 关于 Adam 收敛及其后续发展。国际学习表征会议。获取自 https://openreview.net/forum?id=ryQu7f-RZ Rushing, H., Karl, A., & Wisnowski, J. (2014). *Douglas Montgomery* 编著的《实验设计与分析》：*JMP* 使用补充指南。卡里：SAS 研究所。Winters, P. (1960). 通过指数加权移动平均预测销售。《管理科学》, 6(1): 127-137。Zhu, L., & Laptev, N. (2017). Uber 的时间序列深度置信预测。*arXiv*。获取自 https://arxiv.org/abs/1709.01907