基于改进型人机交互集成深度学习方案的跨设备响应能力评估增强型网页用户界面设计

Shrinivass Arunachalam Balasubramanian*

高级全栈工程师、独立研究员，美国，shrinivassab@gmail.com

在数字时代，用户界面（UI）优化对于提升网络环境中的用户满意度至关重要。然而，现有的用户界面优化模型忽视了跨平台响应能力（CR）评估，从而影响了用户交互效率。因此，本文提出一种通过跨平台响应能力评估实现动态网络用户界面优化的方法，采用有限指数连续状态机（FECSM）和Quokka非线性差分群优化算法（QNDSOA）。首先，收集设计及用户交互相关信息，并对其进行预处理和最小-最大归一化。随后提取基于人机交互（HCI）的特征，并进行用户行为模式分组。同时，利用FECSM进行跨平台响应能力评估。然后，采用所提出的双向门控Luong与Mish循环单元（BiGLMRU）对用户体验（UX）变化类型进行分类，并依据用户界面变化预测指数（UICPI）进行标注。最后，利用新型QNDSOA优化用户界面设计，平均适应度达到98.5632%。完成最优部署后进行反馈监测。

补充关键词与短语：人机交互（HCI）、用户界面（UI）优化、网页开发（WD）、用户体验（UX）建模、预测性用户界面增强（PUIE）、模糊导数加权推理系统（FDWIS）和人工智能（AI）。

1. 引言

当前，网站已成为用户界面的重要平台；在这个以用户为中心的时代，网站架构能够提升用户体验 [1, 2]。因此，网页开发作为人机交互的重要方面，被用于改善用户体验。这包括对用户行为（UB）模式进行估计、创建用户友好型界面，以及测试网站的性能 [3, 4]。屏幕尺寸影响和眼动追踪方法等用户行为模式对改善用户体验大有裨益 [5, 6]。通过开发适老化网站，电子商务等平台能够改善用户体验 [7, 8]。然而，传统技术并未专注于网页开发的 CR 评估 [9, 10]。所提出系统的目标是开发一个以用户行为为核心、配备相应界面的用户友好型网站。因此，本文提出了一种基于 BiGLMRU 优化网站的新模型。

1.1 问题陈述

现有研究的局限性如下：

- ❖ 没有研究关注网页开发中的 CR评估。❖ 没有在 [11]，中聚焦 UB 模式，削弱了用户界面的有效性。❖ 现有研究的用户界面设计效率低下，因为最优用户界面设计没有定期更新。❖ 现有研究未能关注用户界面的改进程度，而这影响了网页开发过程。❖ 在 [12]，中，支持因素未被考虑，进一步降低了有效性。

1.2 目标

所提出的框架的目标如下：

- ❖ 通过采用 FECSM 进行跨平台响应能力评估来开发网络。❖ 为提高效果，使用 HDBSCAN 对 UB 模式进行分组。❖ 通过向 QNDSOA 提供反馈，定期更新最优用户界面设计。

- ❖ 通过使用 BiGLMRU，获得用户界面设计的改进程度。❖ 在最优用户界面设计中，将最短 JavaScript 执行时间、最低错误率和最少内存使用量作为适应度函数，以考虑支持因素。

其余内容安排如下：第2节分析现有研究成果，第3节阐述所提出的方法论，第4节给出结果与讨论，第5节总结本文并展望未来研究方向。

2. 文献综述

Bakaev 等人 [11] 认识到对用户界面视觉感知（VP）进行建模的重要性。在此基础上，利用卷积神经网络（CNN）预测视觉感知。然而，未考虑不同设备和屏幕尺寸下响应性的变化。

Todi 等人 [12] 评估了用于自适应用户界面的强化学习（RL）方法。首先，考虑了影响用户界面的正面和负面因素。因此，自适应用户界面能够调整网页布局并重新组织应用程序菜单。然而，由于未考虑基于最优用户界面设计的支持因素，用户界面的有效性有所降低。

Keselj 等人 [13] 研究了深度学习（DL）在用户界面评估中的应用。在此过程中，卷积神经网络根据用户界面设计和布局等规范确定用户界面的有效性。然而，由于未能实际应用有关用户界面的客观知识，因此未能实现用户满意度。

Muneer 等人 [14] 部署了一种元模型，用于支持跨浏览器 Web 应用程序的兼容性测试（CT）。首先，为覆盖关键配置，他们制定了检查清单，并将其转换为交互流建模语言（IFML）。随后，利用 IFML 生成的测试用例解决了兼容性问题。

Wang 等人 [15] 提出了一种深度学习方法，用于评估人机交互界面中的界面色彩质量。首先，利用 CNN 提取并建模界面图像特征。然而，该方法无法根据用户的即时反馈进行更新，因此用户界面设计仍然效率低下。

3. 提出 ANDBiGL 使用 FECSM 进行跨响应式网络用户界面调优的方法 **MRU**

所提出的工作采用 FECSM 和 QNDSOA 实现了一个用于 CR 网络用户界面优化的智能框架。在图1中，展示了所提出方法的框图。

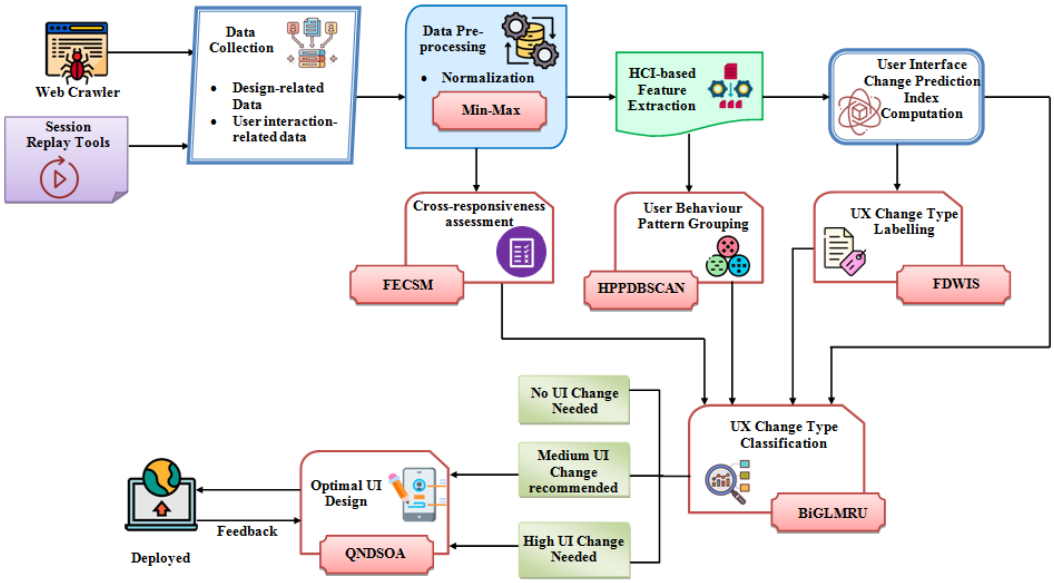

图1：研究方法的结构设计

3.1 数据收集

首先，分别使用网络爬虫和会话回放工具收集设计相关数据（布局、组件排列和响应特性）以及用户交互相关信息（点击、滚动和鼠标移动）。

$$\partial_w = (\partial_1, \partial_2, \ldots\ldots \partial_W) \text{ Where, } w = 1 to W \tag{1}$$

其中，表示收集到的网络数据数量。 W ∂_w

3.2 预处理

接下来对其进行预处理，采用最小-最大归一化将收集的数据标准化到 (0, 1) 范围内。 ∂_w

$$\eta = \frac{\partial_w - \min(\partial_w)}{\max(\partial_w) - \min(\partial_w)} \tag{2}$$

此处，表示预处理后的数据。 η

3.3 基于人机交互的特征提取

从中提取人机交互特征，例如点击模式、滚动行为、鼠标移动和网络状况，从而提升模型的性能。

3.4 用户行为模式分组

接下来，被输入所提出的 HPPDBSCAN 方法中，该方法通过考虑滚动速率和点击深度对用户行为模式进行分组。传统的具有噪声的分层密度聚类（HDBSCAN）能够有效地对密度各异的数据进行分组。然而，HDBSCAN 对聚类参数的选择十分敏感，例如最小簇大小 和最小样本数 。因此，采用持久概率函数（PPF），通过分析数据在多个尺度下基于密度的持久性来确定最优参数。

$$\wp(\gamma_i)=\int_{S_k}^{S_K}|\gamma_i(S)|d''S\rightarrow(\mathrm{X},\mathrm{Y}) \tag{3}$$

其中，表示密度水平，在该密度水平下，簇出现；表示簇消失时的密度水平，表示导数参数。对于每个点，核心点根据最小邻居数计算，该数值由确定。同样，为处理不同的密度，会估算点与点之间的相互可达距离。

$$M_{\mathrm{dis}}(\gamma_1,\gamma_2)=\max\{\mathrm{Cp}(\gamma_1),\mathrm{Cp}(\gamma_2),Z(\gamma_1,\gamma_2)\} \tag{4}$$

其中，表示直接距离值。将指定为边的权重值，即可构建完整的相互可达图。随后，通过以最小距离连接所有点且不形成任何环，生成最小生成树。接下来，按照递增顺序对中的边进行排序，然后逐步移除最长的边，以创建层次结构。同时，通过应用进行树剪枝，以评估簇的稳定性。

$$\lambda(T_{\min})=\tilde{\mathrm{X}}(S_K-S_k) \tag{5}$$

最后，将数据点分配至稳定性最高的簇中。因此，按用户行为模式分组的数据将显示为。

3.5 跨设备响应性评估

相反，CR评估采用所提出的 FECSM 算法，来建模界面响应性如何随时间在不同设备间变化。有限状态机（FSM）能够有效捕捉状态转移

设备配置变化会影响用户体验。然而，有限状态机难以处理连续的状态和状态转移，影响了模型的灵活性。因此，采用指数连续覆盖函数来处理变化过程中的状态转移。

此处，每个状态代表一种响应式用户界面（移动端布局、平板布局和桌面布局）。

$$\mathrm{St}_v = (\mathrm{St}_1, \mathrm{St}_2, \ldots\ldots \mathrm{St}_V) \text{ Where, } v = 1 to V \tag{6}$$

此处,V表示状态数量 。接下来，输St_v（用户触发事件）定义如下，包括鼠标事件、屏幕尺寸变化和触控手势，

$$\mathrm{Ip}_w = \sum_{w=1}^{W} \{\mathrm{Ip}_1, \mathrm{Ip}_2, \ldots \mathrm{Ip}_W\} \tag{7}$$

此处，表示输入数量 。此外，定义状态转移，以反映系统在接收到输入后如何从一个状态转移到另一个状态。在所提出的工作中，使用 ECC 函数 ，通过持续适应动态状态变化来确保状态转移的灵活性。

$$\mathrm{N}(\tau) = 1 - \exp^{-\Im\tau} \tag{8}$$

$$\tau \xrightarrow{\text{transition}} (\mathrm{St}_1, \mathrm{St}_v) \tag{9}$$

此处，定控制参数，表示状态转移。随后，还要确定初始状态和最终状态。接下来，将用户交互记录为日志文件。最后，提取 FSM 跟踪日志，以详细了解哪些状态转移造成了摩擦，以及哪种设备布局的任务成功率更高。CR 评估结果如下所示 。

3.6 用户界面变化预测指数计算

同时，通过考虑 来计算CPI，以根据用户交互偏差表示进行用户界面修改的必要性。

$$\varphi(\gamma_i) = \nu_1 \times E + \nu_2 \times T + \nu_3 \times D + \nu_4 \times C \tag{10}$$

其中，表示权重值，表示错误率，表示任务时间，表示流失率，且表示点击混淆指数。

3.7 用户体验变更类型标注

所提出的方法模糊导数加权推理系统（FDWIS）能够基于以下内容精准标注用户体验变更类型。模糊推理系统（ FIS）具有高透明度。然而，FIS 方法难以捕捉细微的

变化。因此，　在去模糊化过程中采用导数加权平均函数（DWAF）来捕捉细微的　变化，提高模型的精度。

首先，模糊化步骤使用 Sigmoid隶属函数将精确值转换为模糊值（隶属度值）。

$$Q(\varphi)=\frac{1}{1+\exp^{-G(\varphi-J)}}\rightarrow \ddot{\varphi} \tag{11}$$

此处，和分别表示控制参数和斜坡中心。

此处，创建模糊if-then规则根据对用户体验变更类型进行分类。

$$\Re_{\text{rule}}=\begin{cases}\text{If}(\ddot{\varphi}=0.0\,\text{to}\,0.3), & \text{then} \quad \text{Lw}\\ \text{If}(\ddot{\varphi}=0.31\text{to}\,0.6), & \text{then} \quad \text{Md}\\ \text{If}(\ddot{\varphi}>0.6), & \text{then} \quad \text{Hw}\end{cases} \tag{12}$$

接下来，将模糊规则应用于模糊化输入，以将用户体验变更类型标记为低、中和高。

接下来，去模糊化的任务是使用 DWAF 将推理引擎的模糊输出转换为精确值。DWAF 优先处理隶属度发生突变的区域，从而提高精度。

$$\varphi=\frac{\sum_{c=1}^{C}O_c(\Re_{rule})\cdot\left|\frac{\dot{b}\varsigma}{\dot{b}\ddot{\varphi}}\right|\cdot\varsigma}{\sum_{c=1}^{C}O_c(\Re_{rule})\cdot\left|\frac{\dot{b}\varsigma}{\dot{b}\ddot{\varphi}}\right|} \tag{13}$$

其中，表示规则的触发强度，表示偏导数参数，且表示模糊规则的数量。

3.8 用户体验变更类型分类

在此，和作为输入传入所提出的 BiGLMRU，该模型根据标注数据将用户界面需求分为三类：需要进行低程度用户界面变更、建议进行中程度用户界面变更，以及必须进行高程度用户界面变更。双向门控循环单元（BiGRU）能够有效捕捉用户交互的动态变化。然而，BiGRU 难以处理较长的依赖关系。因此，采用 Luong 注意力（LA）函数来保留长期信息。同样，BiGRU 也存在过拟合问题。因此，采用 Mish 激活函数

用于最小化　　过拟合问题，改善梯度流。在图2中，所提出的方法 BiGLMRU 的图示说明　　如下。

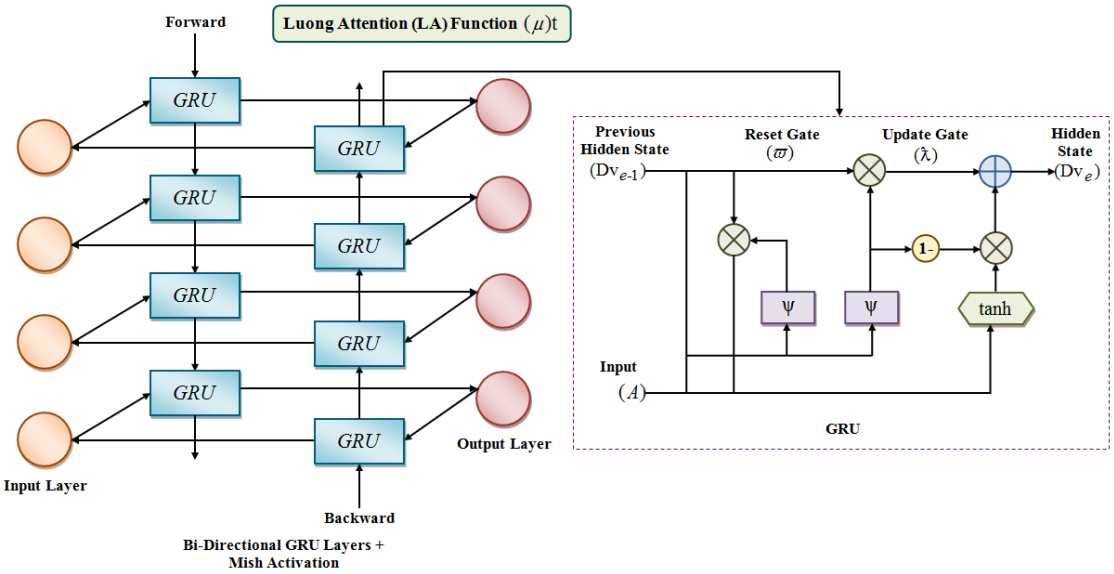

图2：所提出的方法 BiGLMRU 的图示

输入层 容纳输入并将其传递至前向GRU层。

$$A = (\phi_{\nabla}, \mathrm{H}, \varphi) \tag{14}$$

重置门旨在消除隐藏状态中信息量较少的信息。同样，Mish激活函数用于减少过拟合问题，这得益于其梯度稳定性。

$$\varpi = \psi \times ((A, \mathrm{Dv}_{e\text{-}1}) \cdot \mathrm{Nu}) + \mathrm{Xg} \tag{15}$$

$$\psi(\quad) = \quad\quad\quad\

$$\mu = \tanh\left(\mathrm{Nu}\left[\sum \chi \cdot \aleph; \left(A, \mathrm{Dv}_{e\text{-}1}\right)\right]\right) \tag{18}$$

此处，χ说明了softmax函数并描述了概率分数ℵ 接下来，候选隐藏状态根据进行计算，从而保留长序列。

$$\tilde{\mathrm{D}}\mathrm{v}_e = \tanh\left(\varpi \cdot \left(A, \mathrm{Dv}_{e\text{-}1}\right) \cdot \mathrm{Nu}\right) + \mathrm{Xg} \tag{19}$$

最后，隐藏状态通过对前一隐藏状态和候选隐藏状态进行加权平均来计算。

$$\mathrm{Dv}_e = \mu \cdot \left(\left(1 - \lambda\right) \cdot \mathrm{Dv}_{e\text{-}1} + \lambda \cdot \tilde{\mathrm{D}}\mathrm{v}_e\right) \tag{20}$$

此外，输出层通过反向GRU层处理输入。在这里，所提出的方法 BiGLMRU 将用户界面变更需求分类为低、中和高效等级。

$$\hat{\mathrm{v}} = \left(L_{\mathrm{UI}}, M_{\mathrm{UI}}, H_{\mathrm{UI}}\right) \tag{21}$$

在此，v̂建立所提出的方法 BiGLMRU的结果。

所提出的方法 BiGLMRU 的伪代码如下，

输入：ϕ∇和H φ

输出：用户界面变更需求分类

开始

初始化：、ϕ∇和 H φ ψ

对每个输入从 1 开始执行，

确定输入层

$$A = \left(\phi_{\nabla}, \mathrm{H}, \varphi\right)$$

#GRU 层

执行重置门，

ϖ = ψ × · +eA

激活 Mish 函数，

执行更新门 = ψ × · +eAλ

建立 LA 函数

$$\mu = \tanh\left(\mathrm{Nu}\left[\sum \chi \cdot \aleph; \left(A, \mathrm{Dv}_{e\text{-}1}\right)\right]\right)$$

计算 **e** candidate hidden state

估算隐藏状态

$$\mathrm{Dv}_e = \mu \cdot \left(\left(1 - \lambda\right) \cdot \mathrm{Dv}_{e\text{-}1} + \lambda \cdot \tilde{\mathrm{D}}\mathrm{v}_e\right)$$

执行输出层 HMLν = (UI UI UI),

结束循环

返回 $\hat{\nu}$

End

3.9 最优用户界面设计

所提出的 QNDSOA 是根据中等 和高用户界面变更 以优化用户界面设计而建立的 M_{UI} 袋鼠群优化算法（QSOA）H_{UI} 在调整加速度系数等参数方面具有很强的适应性。但是，QSOA 难以确定种群中各成员的位置。因此，采用非线性差异函数（NDF）来反映种群中各成员位置的多样性。

首先，在搜索区域内初始化种群成员。在此，将字体大小、主题模式、字间距和文本对齐等输入视为短尾矮袋鼠（成员）。

$$L_x = \{L_1, L_2, \ldots\ldots L_X\} \text{ Where, } x = 1 to X \tag{22}$$

此处，X 表示种群成员的数量。在所提出的工作中，将最小 L_x JavaScript 执行时间、最低错误率和最小内存使用量视为适应度值，以选择最佳领导者。随后，根据更新成员位置和干旱程度。所提出的工作引入 NDF，以涵盖整个种群中位置的多样性。

$$\text{Dh}_{新} = \frac{\text{hmTm}}{\text{Dh0.8}} + \iota \cdot \alpha \tag{23}$$

$$\alpha = \exp^{-O\hbar}\left(L_x^{Best} - L_x\right) \tag{24}$$

$$= \quad + \quad * \sigma_{新} \text{Dh}^{new} \tag{25}$$

其中，表示更新后的干旱程度，表示更新后的成员位置，表示温度（平衡参数），表示湿度（探索力），表示氮比（解的质量），表示领导者与成员之间的权重，表示自适应参数，表示时间界限，而表示领导者与短尾矮袋鼠之间的位置差异。接下来，适应度会不断迭代更新，直至收敛。下面给出了QNDSOA 的伪代码，

输入：Web 组件

输出：最优 Web 用户界面 （ UI ）

O

开始

初始化，L_x, L_x^{Best} 和 Dh σ

对于 1 到每个成员，执行

执行种群初始化

LLLL 、 = 1 2

根据适应度值选择领导者 L_x^{Best}

更新干旱，

$$Dh_{新} = \frac{\left(\frac{hmTm}{Dh0.8}\right)}{\left(+\right)} + \iota \cdot \alpha$$

应用 NDF $\alpha = \quad -Oh \quad -\exp$

更新成员位置

$= \quad + \quad * \sigma_{新} Dh^{new}$

重复直至收敛

结束循环

返回 O_{UI}

End

所提出的方法 QNDSOA 通过平衡不同设备上的响应性来优化用户界面布局。生成优化后的用户界面设计后，便将其部署到实际运行的网络环境中。系统会持续监测用户与新用户界面的交互，并据此反馈对模型进行微调。

4. 结果与讨论

实验研究旨在验证所提出的工作在 Python 平台上的性能。

4.1 数据集描述

通过网络爬虫和会话回放工具实时收集与设计相关的信息以及用户交互相关信息，以评估所提出的方法。全部数据中，80% 和 20% 分别用于训练和测试。

4.2 研究方法的性能评估

所提出的工作与多种现有算法进行了性能比较。

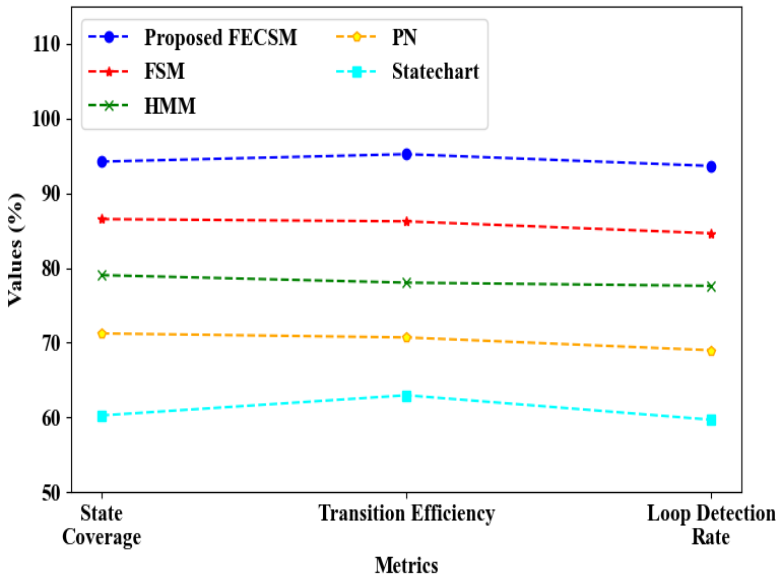

图 3：CR评估的实证分析

表 1：所提出的 FECSM 的性能验证

算法	状态覆盖率（%）	转换效率（%）	循环检测率（%）
所提出的方法 FECSM	94.2356	95.2312	93.6532
FSM	86.5402	86.2356	84.6375
HMM	79.0364	78.0326	77.6023
PN	71.2341	70.6982	68.9782
状态图	60.2584	62.9584	59.6803

图3和表1对比了所提出的FECSM与现有的有限状态机、隐马尔可夫模型（HMM）、Petri网（PN）和状态图。所提出的FECSM的状态覆盖率、转换效率和循环检测率分别达到94.2356%、95.2312%和93.6532%。相比之下，传统有限状态机的状态覆盖率、转换效率和循环检测率分别为86.5402%、86.2356%和84.6375%，表明其适应性有限。在此，ECC的引入有助于提升所提出的工作在CR评估中的性能。

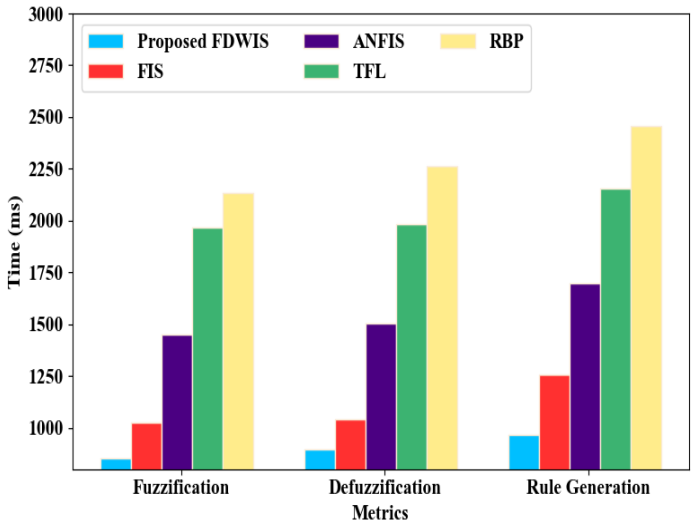

图4：用户体验变更标注的性能评估

表2：所提出的方法 FECSM 的数值研究

技术	模糊化时间（毫秒）	去模糊化时间（毫秒）	规则生成时间（毫秒）
所提出的方法 FDWIS	855	897	963
FIS	1024	1042	1255
自适应神经模糊推理系统	1450	1501	1698
TFL	1964	1985	2157
RBP	2135	2264	2455

在图4和表2中，将所提出的 FDWIS 与当前的 FIS、自适应神经模糊推理系统（ANFIS）、梯形模糊逻辑（TFL）以及基于规则的预测（RBP）进行比较，以展示该模型在用户体验变更标注方面的优势。采用基于DWAF的去模糊化提升了标注效果。所提出的 FDWIS 的模糊化时间、去模糊化时间和规则生成时间分别为855ms、897ms 和 963ms。然而，传统技术耗时最长。因此，FDWIS 的优势得到了证实。

表3：用户体验变更类型分类的比较评估

方法	准确率（%）	精度（%）	召回率 (%)	F值 (%)	敏感度 (%)	特异度 (%)
所提出的方法 BiGLMRU	99.2315	98.0745	98.1265	98.1005	98.1265	98.0745
BiGRU	93.5489	90.4568	91.6544	91.0556	91.6544	90.4568
LSTM	88.9746	87.4571	88.0267	87.7419	88.0267	87.4571
RNN	84.3922	81.9655	80.9472	81.4563	80.9472	81.9655
DLNN	77.3586	76.3204	77.5543	76.9373	77.5543	76.3204

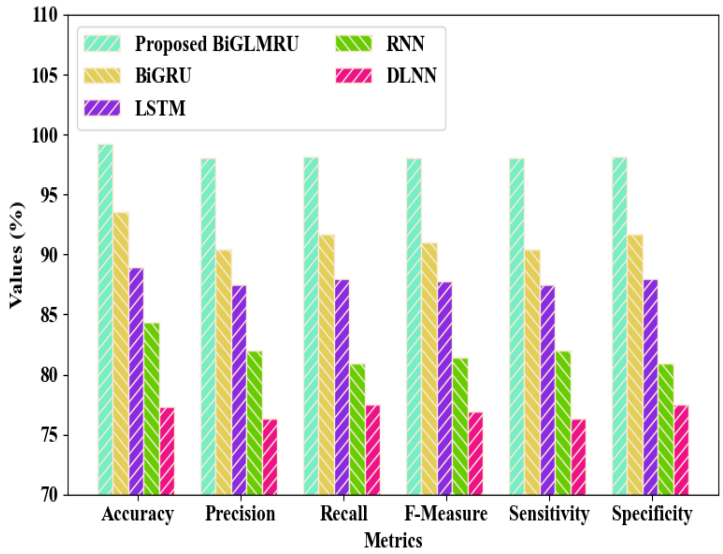

图5：所提出的方法 BiGLMRU 的性能评估

在图5和表3中，将所提出的 BiGLMRU 与当前的 BiGRU、长短期记忆（LSTM）、循环神经网络（RNN）和深度学习神经网络（DLNN）进行比较。就准确率、精度、召回率、F值、敏感度和特异度而言，BiGLMRU 分别达到 99.2315%、98.0745%、98.1265%、98.1005%、98.1265% 和 98.0745%；而当前技术分别达到 86.0685%、84.0499%、84.5456%、84.2977%、84.5456% 和 84.0499%。现有研究的分类性能较差。不过，BiGLMRU 使用 Mish激活函数来缓解过拟合问题，进一步提升了模型的优势。

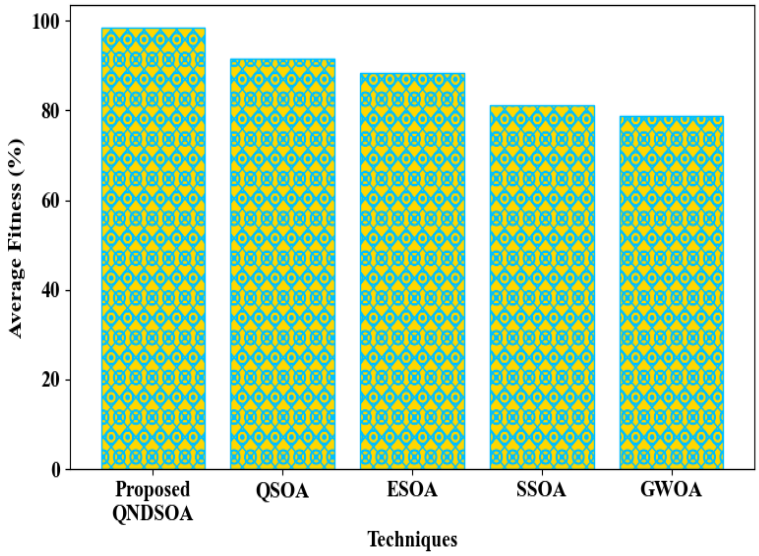

图6：平均适应度分析

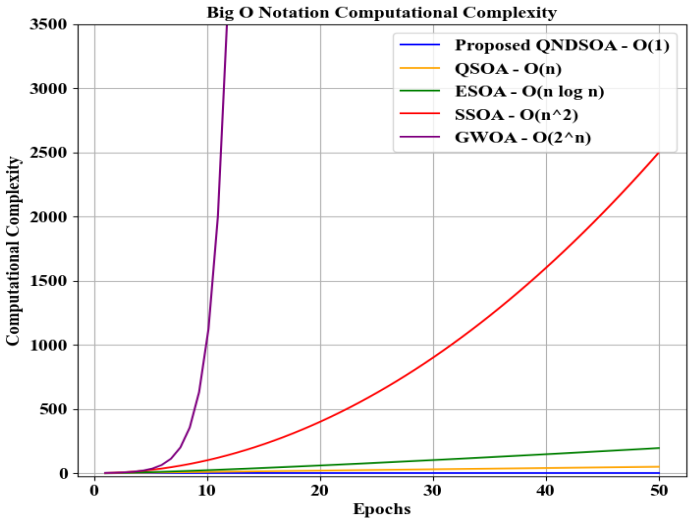

图 7：用户界面优化的计算复杂度分析

图 6 和图 7 对所提出的 QNDSOA 与当前主流的 QSOA、白鹭群优化算法（ESOA）、水蚤群优化算法（SSOA）以及灰狼优化算法（GWOA）进行了性能比较。所提出的 QNDSOA 的平均适应度达到 98.5632%，而传统 GWOA 为 78.6594%。此外，由于采用基于 NDF 的位置更新方法，所提出的工作在迭代轮数变化时复杂度较低。因此，QNDSOA 的性能更优。

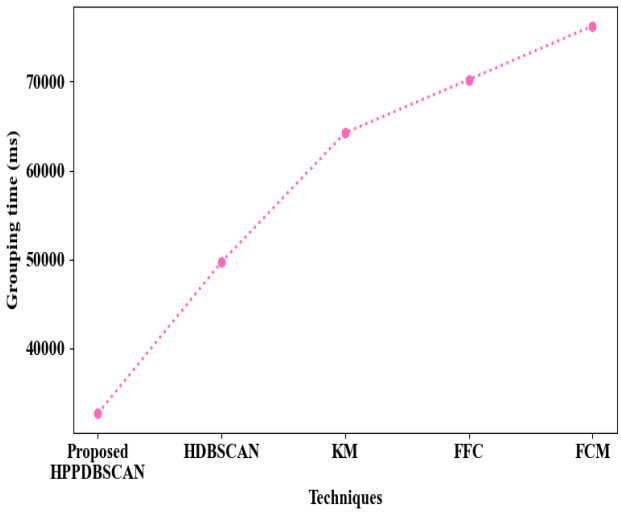

图 8：用户行为模式分组的性能评估

在图 8 中，就分组时间而言，所提出的 HPPDBSCAN 与现有的 HDBSCAN、K均值（KM）、最远优先聚类（FFC）和模糊C均值（FCM）进行了比较。所提出的 HPPDBSCAN

耗时 32654 毫秒 完成分组，而现有的 HDBSCAN 分组耗时为 49687ms。因此 re,
所提出的方法 由于参数选择得当，所提出的工作具有较低的时间复杂度。

4.3 所提出的工作的比较验证

研究方法的比 较分析旨在展示模型的优 显著性。

表4：比较验证

作者姓名	目标领域	方法	优点	挑战
Ma [16]	计算机网络界面优化	BPNN	更快的页面加载	脚本执行延迟
Wang [17]	智能布局适应于网页设计	NCMF	更高的用户参与度	动态内容不稳定性
Kikuchi 等人 [18]	增强型网页页面布局优化	基于优化的分层布局 mode	改进的响应式设计	布局偏移问题
Martin 等人 [19]	个性化网页用户界面适应	情境适应型感知方案	更好的灵活性和适应性	过度响应元素
Xu & Wang [20]	交互式网站搜索界面设计	凹凸纹理映射算法	增强型网页无障碍性	较低的适应性
所提出的工作	增强型网页用户界面设计通过 CR评估使用先进的人机交互	FECSM 和 QNDSOA	改进的跨兼容性和自适应布局设计	所提出的工作严重依赖用户界面优化，而非可解释性

在表4中，将所提出的工作的性能与若干相关研究进行了比较。所提出的 FECSM 和 QNDSOA 算法通过 CR 评估，帮助提升了网络环境中的用户体验。同样，为了优化计算机网络界面，已有研究采用了反向传播神经网络（BPNN）（Ma，2022年）和非负卷积矩阵分解（NCMF）（Wang，2022年）。然而，已有工作存在适应性问题和计算开销。因此，所提出的工作以较低的复杂度实现了自适应布局设计。

5 结论

本文通过 CR 评估，采用改进的人机交互集成 FECSM 和 QNDSOA 方法，提出了一种增强型网络用户界面设计。所提出的 FECSM 详细揭示了不同屏幕尺寸下的状态转移，状态覆盖率达到 94.2356%。同样，新颖的 QNDSOA 显著优化了网络用户界面，平均适应度达到 98.5632%。此外，通过持续反馈监测确保模型的可信度。尽管如此，所提出的工作主要侧重于优化用户界面设计，而非解释。

未来展望：因此，未来本研究将着重考虑可解释人工智能和认知负荷因素，以提高用户界面增强流程的可靠性和可信度。

参考文献

[1] Lun Liu, Dai Zetian, Tan Wee Hoe, Xue Juan, Du Jiaxin, and Wang Fulai. 2024. 交互式网站上影响用户意图的因素：来自技术接受模型的见解。IEEE Access 12 (2024年8月), 122735–122756. https://doi.org/10.1109/ACCESS.2024.3437418[2] Lima, Adriano Luiz de Souza, and Christiane Gresse von Wangenheim. 2021. 用户界面的视觉美学评估：十年系统性映射。International Int. J. Hum.-Comput. Interact. 38, 2 (2021年1月), 144–164. https://doi.org/10.1080/10447318.2021.1926118**[3]** Alao Olujimi Daniel, Amarachi Priscilla Ezihe, Ruth Chinkata Amanze, Oluwakemi Kuyoro Shade, and Adewale Olanrewaju Adebayo. 2022. 面向用户的用户体验设计思维方法：大学信息管理系统设计。Ing. Syst. Inf. 27, 4 (2022年8月): 577. https://doi.org/10.18280/isi.270407[4] Hossain Md Tutul, Rakib Hassan, Mahfida Amjad, and Md Abdur Rahman. 2021. 网页性能分析：对孟加拉国电子商务网站的实证分析。In Kuma https: procs.2025.04.567[6] Ball Linden J. and Beth H. Richardson. 2023. 面向用户体验评估的眼动追踪与生理测量。人机交互中的用户体验方法与工具 (2024年8月), 1–31. https://www.taylorfrancis.com/chapters/edit/10.1201/9781003495161-9/eye-tracking-physiological-measurements-ux-evaluation-linden-ball-beth-richardson[7] Ye Junnan, Yueting Han, Wenhao Li, and Chaoxiang Yang. 2025. 面向老年人的生鲜电子商务产品界面视觉选择性注意分析。Appl. Sci. (Switz.) 15, 8 (2025年4月), 1–26. https://doi.org/10.3390/app15084470[8] Pathak, Bhavesh, Sandeep Mamloda, and Manthan Patel. 2025. 响应式电子商务网站。Research Gate (2025年3月), 1–14. https://www.researchgate.net/publication/390151475_Responsive_E-Commerce_Website[9] Alti Adel and Abderrahim Lakehal. 2025. Al 借助生成式人工智能和模型驱动开发革新电子商务用户体验。Future Internet 17, 4 (2025年4月), 1–34. https://doi.org/10.3390/fi17040180[10] Khamaj Abdulrahman and Abdulelah M. Ali. 2024. 利用强化学习适应用户体验：基于实时用户行为分析实现界面个性化。Alex. Eng. J. 95 (2024年5月), 164–173. https://doi.org/10.1016/j.aej.2024.03.045[11] Bakaev Maxim, Sebastian Heil, and Martin Gaedke. (2023). 用户界面视觉感知建模中特征具有合理的有效性。Big Data Cogn. Comput. 7, 1 (2023年2月), 1–17. https://doi.org/10.3390/bdcc7010030[12] Todi Kashyap, Gilles Bailly, Luis Leiva, and Antti Oulasvirta. 2021. 基于模型的强化学习适配用户界面。载于《2021年计算机系统中的人为因素国际会议论文集》（CHI '21）。计算机协会，纽约州纽约市，美国，1–13. https://dl.acm.org/doi/abs/10.1145/3411764.3445497[13] Kesel av Zubrinic, and Zeljka Car. 2022. 深度学习在用户界面评估中的应用。Sensors 22, 23 (2022年11月), 1–17. https://doi.org/10.3390/s22239336[14] Muneer Mishal, Uzair Rasheed, and Muhammad Majid Hussain. 2025. 支持跨浏览器网页应用兼容性测试的元模型。载于《2025年第六届计算科学进展国际会议》（ICACS '25），IEEE，巴基斯坦拉合尔，1–8。https://ieeexplore.ieee.org/abstract/document/10937860/[15] Wang Shixiao, Runsheng Zhang, Junliang Du, Ran Hao, and Jiacheng Hu. 2025. 人机交互中的界面色彩质量评估深度学习方法。ArXiv (2025年2月), 1–5. https://arxiv.org/pdf/2502.09914[16] Ma Yan. 2022. 基于BP神经网络算法和多媒体的计算机网页界面优化。Comput. Intell. Neurosci. 2022, 1 (2022年5月), 1–8. https://doi.org/10.1155/2022/6213718[17] . 2022. 机器视觉下人工智能对网页设计视觉元素的影响。Comput. Intell. Neurosci. 2022, 2 (2022年5月), 1–13. https://doi.org/10.1155/2022/4328400[18] Kikuchi Kotaro, Mayu Otani, Kota Yamaguchi, and Edgar Simo-Serra. 2021. 面向网页布局优化的视觉包含建模。Comput. Graph. Forum. 40, 7 (2021年10月), 33–44. https://doi.org/10.1111/cgf.14399[19] Märtin Christian, Bärbel Christine Bissinger, and Pietro Asta. 2023. 优化数字客户旅程——利用情绪、用户画像和情境实现个性化用户界面适配，从而改善用户体验。J. Consum. Behav. 22, 5 (2023年9月), 1050– 1061. https://doi.org/10.1002/cb.1964

[20] Xu Zhen 和 Shan Wang. 2022年. 基于视觉传达的个性化网站搜索界面交互设计. Comput. Intell. Neurosci. 2022年 (2022年5月), 1–11. https://doi.org/10.1155/2022/2125506