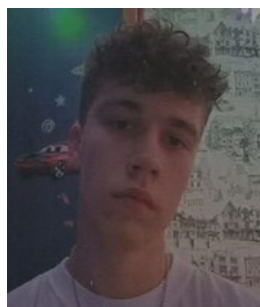


УДК 004.8

ПРИМЕНЕНИЕ МАШИННОГО ОБУЧЕНИЯ ДЛЯ ПЕРСОНАЛИЗАЦИИ АДАПТИВНОГО ДИЗАЙНА



Т.А. Васяева
Декан факультета
информационных систем и
технологий ДонНТУ,
кандидат технических наук,
доцент
tanechka.vasyaeva@yandex.ru



М.А. Рябко
Магистрант кафедры
АСУ, ДонНТУ
maxryabko2706@yandex.ru

Т.А. Васяева

Кандидат технических наук, доцент, декан факультета «Информационные системы и технологии», ФГБОУ ВО «Донецкий национальный технический университет». Область научных интересов связана с исследованием и разработкой методов искусственного интеллекта, организацией учебного и научно-исследовательского процессов в техническом университете.

М.А. Рябко

Магистрант кафедры автоматизированных систем управления, ФГБОУ ВО «Донецкий национальный технический университет». Область научных интересов связана с исследованием и разработкой методов адаптивного и мобильного дизайна веб-сайтов.

Аннотация. В современном цифровом мире персонализация играет ключевую роль в создании эффективного пользовательского опыта. Использование машинного обучения (МО) позволяет значительно улучшить адаптивный дизайн веб-сайтов, обеспечивая более релевантный контент и интерфейс для пользователей. В данной статье рассматриваются проблемы традиционного адаптивного дизайна, методы персонализации с использованием МО, а также предлагается практическая реализация данного подхода.

Ключевые слова: Адаптивность, машинное обучение, веб-сайт, генетический алгоритм, нейронные сети.

Введение. В последние годы персонализация цифрового контента и пользовательского опыта становится одним из ключевых направлений в развитии веб-технологий. С ростом количества интернет-пользователей и разнообразия устройств, с которых осуществляется доступ к веб-сайтам, традиционные методы адаптивного дизайна перестают быть эффективными. Они учитывают лишь технические характеристики устройства, такие как размер экрана или разрешение дисплея, но не принимают во внимание индивидуальные предпочтения пользователя. Это приводит к стандартному, одинаковому для всех опыту взаимодействия, который не всегда соответствует ожиданиям конкретного пользователя.

Применение машинного обучения (МО) позволяет решить эту проблему за счёт анализа данных о поведении пользователей и предсказания их потребностей. Алгоритмы МО способны выявлять скрытые закономерности в пользовательских данных, адаптируя интерфейс веб-сайтов в режиме реального времени. Это делает возможным создание

персонализированных решений, которые не только улучшают пользовательский опыт, но и повышают показатели вовлечённости, конверсии и удовлетворённости аудитории.

Цель данной работы. Исследовать возможности применения МО для персонализации адаптивного дизайна веб-сайтов, проанализировать существующие методы и предложить модель для улучшения пользовательского опыта. Актуальность темы обусловлена тем, что персонализация интерфейса становится конкурентным преимуществом для компаний, работающих в цифровом пространстве. Использование методов МО в адаптивном дизайне позволяет веб-разработчикам и владельцам интернет-ресурсов предоставлять контент, максимально соответствующий потребностям пользователей, что ведёт к улучшению взаимодействия с сайтом и увеличению коммерческой эффективности.

Постановка задачи. Развитие цифровых технологий и рост количества интернет-ресурсов ставят перед разработчиками веб-дизайна важную задачу: создать интерфейсы, которые не только адаптируются под технические параметры устройства, но и учитывают индивидуальные предпочтения пользователя. Традиционные методы адаптивного дизайна фокусируются в основном на статических параметрах, таких как разрешение экрана, но не принимают во внимание поведенческие и контекстные факторы.

Основные задачи исследования включают:

- 1 Анализ традиционных методов адаптивного дизайна и их ограничений в контексте персонализации.
- 2 Исследование методов машинного обучения, применяемых для персонализации пользовательского опыта.
- 3 Разработку модели персонализированного адаптивного дизайна с применением МО.
- 4 Оценку эффективности предложенной модели на основе ключевых метрик, таких как вовлеченность пользователей, среднее время на сайте, коэффициент конверсии.
- 5 Решение данных задач позволит предложить инновационные методы персонализации интерфейсов, улучшив качество взаимодействия пользователей с цифровыми платформами.

Описание проблемы. Современный адаптивный дизайн сталкивается с рядом проблем, связанных с обеспечением персонализированного пользовательского опыта. Традиционные методы адаптации интерфейсов учитывают лишь технические параметры устройств, игнорируя индивидуальные предпочтения и поведение пользователей. Это приводит к созданию унифицированных интерфейсов, не всегда соответствующих ожиданиям конкретных пользователей.

Основные проблемы адаптивного дизайна:

- 1 Ограниченная персонализация. Традиционные подходы к адаптивному дизайну фокусируются на технических характеристиках устройств, таких как размер экрана и разрешение, не учитывая индивидуальные предпочтения пользователей. Это приводит к созданию однотипных интерфейсов, не отражающих уникальные потребности каждого пользователя.
- 2 Сложности в отображении контента на разных устройствах. Адаптация сложных элементов, таких как таблицы данных, на небольших экранах представляет значительные трудности. Это может ухудшать пользовательский опыт и снижать эффективность взаимодействия с веб-ресурсом.
- 3 Проблемы навигации и визуализации. На небольших экранах навигационные элементы и визуальные компоненты могут отображаться некорректно, что затрудняет доступ к информации и снижает удовлетворенность пользователей.

Роль машинного обучения в решении проблем адаптивного дизайна. Интеграция искусственного интеллекта (ИИ) и машинного обучения (МО) в процессы веб-дизайна

открывает новые возможности для персонализации и адаптации интерфейсов. Такие как, например:

1 Анализ пользовательского поведения. Алгоритмы МО могут обрабатывать большие объемы данных о поведении пользователей, выявляя скрытые закономерности и предпочтения. Это позволяет создавать интерфейсы, адаптированные под индивидуальные потребности, улучшая пользовательский опыт.

2 Автоматизация дизайна. ИИ способен автоматически генерировать дизайн-решения на основе анализа предпочтений пользователей, что ускоряет процесс разработки и обеспечивает более точное соответствие ожиданиям аудитории.

3 Персонализация контента. Использование МО позволяет предлагать пользователям контент, наиболее соответствующий их интересам, повышая вовлеченность и удовлетворенность.

Варианты решения проблемы. В процессе персонализации адаптивного дизайна могут использоваться различные методы машинного обучения. Рассмотрим подробнее основные подходы и их применение:

1 Кластеризация пользователей

Кластеризация позволяет группировать пользователей на основе их поведения, предпочтений и демографических характеристик. Следует рассмотреть наиболее популярные алгоритмы:

K-means – простой и эффективный метод, разделяющий пользователей на группы по схожести их данных. DBSCAN – полезен для выделения аномалий и сегментирования пользователей без необходимости задавать количество кластеров заранее. Иерархическая кластеризация – строит древовидную структуру взаимосвязей между пользователями.

2 Коллаборативная фильтрация

Метод широко используется в рекомендательных системах. Разделяется на два типа:

- пользовательская коллаборативная фильтрация – рекомендации на основе сходства между пользователями;
- контентная коллаборативная фильтрация – анализирует предпочтения пользователей на основе характеристик контента.

Применение: подбор динамических элементов интерфейса в зависимости от предпочтений пользователей, персонализированные рекомендации товаров или контента.

3 Нейронные сети и глубокое обучение

Глубокие нейронные сети (DNN) позволяют анализировать сложные паттерны поведения пользователей. Основные архитектуры:

- рекуррентные нейронные сети (RNN) – используются для анализа последовательностей действий пользователя;
- сверточные нейронные сети (CNN) – применяются в анализе изображений, полезны для персонализации графического контента;
- генеративно-состязательные сети (GAN) – позволяют генерировать персонализированные элементы дизайна.

4 Подходы на основе градиентного бустинга

Такие алгоритмы, как XGBoost, LightGBM, CatBoost, эффективно работают с табличными данными и позволяют находить скрытые зависимости в пользовательском поведении. Они применяются для прогнозирования конверсий и анализа влияния изменений в дизайне.

Реализация задачи исследования. Для демонстрации предлагаемого подхода была разработана модель на основе нейронной сети, которая анализирует данные о поведении пользователей и адаптирует элементы дизайна веб-сайта в реальном времени. В качестве примера использовались следующие технологии: Python, TensorFlow, Scikit-learn. Данные технологии использовались для реализации модели машинного обучения.

В рамках данной работы был реализован генетический алгоритм для оптимизации параметров персонализации адаптивного дизайна. Также разработана нейросетевая модель для предсказания вероятности конверсии пользователя на основе его поведения на сайте. Хотя интеграция в реальный веб-интерфейс не была осуществлена, данный подход демонстрирует возможность интеллектуального подбора параметров дизайна. Разберем подробнее разработки в рамках данного научного исследования.

1 Генетический алгоритм для персонализации дизайна

Генетический алгоритм (ГА) использовался для оптимизации параметров интерфейса, таких как размер шрифтов, цветовая гамма, расположение элементов и структура навигации. Основные этапы его работы:

Генерация начальной популяции – случайный набор конфигураций дизайна.

Оценка приспособленности (fitness-функция) – каждая конфигурация оценивается по ряду метрик, например, удобочитаемость, визуальное восприятие, предполагаемая вовлечённость пользователя.

Скрещивание и мутации – комбинация лучших конфигураций с небольшими случайными изменениями.

Выбор лучших решений – отбор вариантов, обеспечивающих наилучший пользовательский опыт.

2 Нейросетевая модель для предсказания конверсии

Параллельно с генетическим алгоритмом была создана нейросетевая модель, обученная на данных о пользовательском поведении. Она предсказывает вероятность конверсии (совершения целевого действия) на основе трёх параметров: времени на странице, количества кликов и глубины прокрутки.

Процесс работы модели:

- сбор и обработка данных;
- генерация случайного набора данных (1000 записей) с имитацией пользовательского поведения;
- стандартизация признаков (время на странице, количество кликов, глубина прокрутки);
- разделение данных на обучающую (80%) и тестовую (20%) выборки;
- создание и обучение нейронной сети;
- многослойная нейросеть с двумя скрытыми слоями (16 и 8 нейронов) и сигмоидным выходным слоем;
- оптимизатор Adam, функция потерь – бинарная кросс-энтропия;
- обучение модели на 20 эпохах с батчами по 32 записи;
- оценка точности и предсказания. Средняя точность модели на тестовой выборке – около 0.7.

Разберем подробнее одну из важнейших функций в рамках данной работы. Функция `predict_user_experience` принимает поведенческие данные пользователя и выдаёт рекомендацию: персонализированный или стандартный дизайн. Генетический алгоритм использует предсказания нейросети для выбора наиболее эффективных параметров дизайна. Например, если модель указывает, что пользователи с высоким временем на странице лучше конвертируются при определённой цветовой схеме, алгоритм закрепляет это правило.

Рассмотрим некоторые этапы в разрезе разработанного программного кода. Как было сказано ранее, для разработки использовался язык программирования Python в связке с фреймворками TensorFlow и Scikit-learn.

1 Сбор данных о поведении пользователей (время на странице, клики, взаимодействия).

Программная реализация данного пункта представлена на рисунке ниже:

```
# Генерация случайных данных
# Данные моделируют поведение пользователей на сайте: время на странице, клики, глубину прокрутки и факт конверсии.
np.random.seed(42)
n_samples = 1000
data = {
    "time_on_page": np.random.normal(60, 15, n_samples), # Среднее время на странице
    "clicks": np.random.randint(1, 10, n_samples), # Количество кликов пользователя
    "scroll_depth": np.random.uniform(0.2, 1.0, n_samples), # Глубина прокрутки
    "conversion": np.random.randint(0, 2, n_samples) # Факт конверсии (0 или 1)
}
```

Рисунок 1. Функция сбора данных о поведении пользователей

2 Обучение модели на исторических данных.

В данном этапе работы, программа разделяет данные на признаки и целевую переменную, затем данные нормализуются для ускорения сходимости модели и повышения точности предсказаний. В ходе работы программы, после нескольких запусков данного кода, средняя точность предсказаний была 0,7, что является приемлемым результатом для конкретного задания. После нормализации данных, они разделяются на обучающую и тестовую выборки. Затем, создается многослойная нейронная сеть для предсказания вероятности конверсии пользователя. Для компиляции модели используется оптимизатор Adam, а в качестве функции потерь используется бинарная кросс-энтропия. Следующим этапом работы является обучение модели на тренировочных данных, после чего формируется оценка точности модели на тестовой выборке.

```
# Разделение данных на признаки (X) и целевую переменную (y)
X = df.drop("conversion", axis=1)
y = df["conversion"]

# Нормализация данных для ускорения сходимости модели и повышения точности предсказаний
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Разделение выборки на обучающую (80%) и тестовую (20%)
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)

# Создание многослойной нейронной сети для предсказания вероятности конверсии пользователя
model = keras.Sequential([
    keras.layers.Dense(16, activation='relu', input_shape=(X_train.shape[1],)), # Первый скрытый слой
    keras.layers.Dense(8, activation='relu'), # Второй скрытый слой
    keras.layers.Dense(1, activation='sigmoid') # Выходной слой с функцией активации сигмоиды
])

# Компиляция модели с использованием Adam-оптимизатора и бинарной кросс-энтропии в качестве функции потерь
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# Обучение модели на тренировочных данных
model.fit(X_train, y_train, epochs=20, batch_size=32, validation_data=(X_test, y_test))

# Оценка точности обученной модели на тестовой выборке
loss, accuracy = model.evaluate(X_test, y_test)
print(f"Модель обучена с точностью: {accuracy:.2f}")
```

Рисунок 2. Функции для обучения модели

3 Предсказание предпочтений пользователей в реальном времени.

Программная реализация данного пункта представлена на рисунке ниже:

```
# Функция для предсказания вероятности конверсии пользователя
# На основе его поведения на сайте (время на странице, количество кликов и глубина прокрутки)
def predict_user_experience(time_on_page, clicks, scroll_depth):
    user_data = np.array([[time_on_page, clicks, scroll_depth]])
    user_data_scaled = scaler.transform(user_data) # Применяем нормализацию
    prediction = model.predict(user_data_scaled)[0, 0] # Получаем предсказание модели
    return "Персонализированный дизайн активирован" if prediction > 0.5 else "Обычный дизайн"
```

Рисунок 3. Функция предсказания предпочтений пользователя

В результате работы генетического алгоритма, описанного выше, разработчик получает вариант дизайна сайта, подходящий конкретному пользователю с определенными параметрами. Так, например, на данном этапе работы будет предложено два вида дизайна веб-сайта, а именно – обычный дизайн и персонализированный дизайн. На более продвинутой стадии разработки проекта возможно автоматическое перестроение дизайна веб-сайта под предпочтения конкретного пользователя.

В перспективе возможно расширение возможностей персонализации за счёт:

- интеграции с технологиями дополненной реальности (AR) для адаптации интерфейсов;
- улучшения моделей за счёт мультимодального обучения;
- использования edge computing для персонализации без передачи данных в облако.

Заключение. Использование методов машинного обучения позволяет значительно повысить персонализацию адаптивного дизайна веб-сайтов, улучшая взаимодействие пользователей с ресурсом. В будущем возможно дальнейшее развитие данной технологии с учетом расширенного анализа данных и более сложных моделей нейронных сетей.

Список литературы

- [1] Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.
- [2] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.
- [3] Murphy, K. P. (2012). Machine Learning: A Probabilistic Perspective. MIT Press.
- [4] Chen, J., & Zhang, C. (2021). Personalized Web Experience using Deep Learning. IEEE Transactions on Knowledge and Data Engineering.

Авторский вклад

Васяева Татьяна Александровна – руководство исследованием по применению машинного обучения для разработки адаптивных веб-сайтов.

Рябко Максим Алексеевич – постановка задачи исследования, изучение задачи исследования, разработка решения проблемы, тестирование работы программы и анализ полученных результатов.

APPLYING MACHINE LEARNING TO PERSONALIZE RESPONSIVE DESIGN

T.A. Vasiaeva

*Dean of the Faculty of
Information Systems and
Technologies of DNTU, PhD of
Technical sciences, Associate
Professor*

M.A. Riabko

*Graduate student of the
Department of Automated Control
Systems, DNTU*

Abstract. In today's digital world, personalization plays a key role in creating an effective user experience. The use of machine learning (ML) can significantly improve the responsive design of websites, providing more relevant content and user experience. This article examines the problems of traditional adaptive design, personalization methods using ML, and also suggests a practical implementation of this approach.

Keywords: Adaptivity, machine learning, website, genetic algorithm, neural networks.