

ски распознавать аномалии, но и предлагать интерпретации, которые раньше требовали участия высококвалифицированных специалистов [1, 2]. Искусственный интеллект не заменяет врача, но становится мощным инструментом в его руках.

Особенность предложенного подхода заключается в использовании многослойных архитектур глубокого обучения, которые обучаются на обширных наборах данных, включающих сотни тысяч изображений. Такой подход обеспечивает точное разделение нормальных и патологических состояний, а также автоматическую классификацию заболеваний.

Ключевым преимуществом является сокращение времени диагностики и минимизация человеческого фактора при интерпретации изображений. Однако перед внедрением таких систем в клиническую практику необходимо преодолеть вызовы, связанные с качеством данных, а также соблюдением этических норм при использовании искусственного интеллекта в медицине.

Результаты исследований демонстрируют, что интеграция нейронных сетей в медицинскую диагностику способна не только улучшить качество диагностики, но и открыть новые горизонты в персонализированной медицине.

Литература

1 Ian Goodfellow. Deep Learning / Ian Goodfellow, Yoshua Bengio, Aaron Courville. – MIT Press, 2016. – 800 p.

2 Zeiler M. D. Visualizing and Understanding Convolutional Networks / M. D. Zeiler, R. Fergus // Lecture Notes in Computer Science. – Vol. 8689. – 2014.

Н. С. Михаленко, Е. А. Ружицкая
(ГГУ имени Ф. Скорины, Гомель)

ИСПОЛЬЗОВАНИЕ ТОКЕН-АУТЕНТИФИКАЦИИ ПРИ РАЗРАБОТКЕ ПРИЛОЖЕНИЙ В DJANGO REST FRAMEWORK

Токен-аутентификация – это один из способов проверки подлинности пользователей, предоставляемый Django Rest

Framework (DRF). В этой модели каждому пользователю создается уникальный токен, который клиент передает вместе с каждым запросом для подтверждения своей аутентификации. Токен – это уникальная строка, которая используется для идентификации пользователя. В контексте DRF токен выдается после успешной аутентификации пользователя, и каждый последующий запрос от клиента должен содержать этот токен для доступа к защищенным ресурсам.

Алгоритм работы токена аутентификации:

1 Клиент отправляет запрос на сервер с учетными данными пользователя (например, имя пользователя и пароль).

2 Сервер проверяет учетные данные, и если они верны, создает или возвращает существующий токен для этого пользователя.

3 Клиент сохраняет токен и использует его в каждом последующем запросе, добавляя его в заголовок Authorization.

4 Сервер проверяет токен и разрешает или запрещает доступ к запрашиваемому ресурсу.

Для создания токенов используется модель Token из модуля `rest_framework.authtoken.models`. Токены могут создаваться автоматически при регистрации пользователя с помощью сигналов Django или вручную через административную панель или консоль. Для автоматической генерации токена новому пользователю используется сигнал `post_save`, который связывает создание токена с добавлением записи в таблицу пользователей.

Для повышения безопасности можно расширить базовую функциональность токенов. Например, добавить срок действия токена, чтобы автоматически аннулировать старые токены через определенный период. Токен-аутентификация имеет свои преимущества и ограничения. Она проста в реализации, удобна для RESTful API и легко интегрируется с различными типами клиентов. Однако хранение токенов на клиентской стороне требует дополнительных мер безопасности, таких как шифрование.