

служит отправной точкой, позволяющей найти тот самый баланс, о котором мечтает каждый сознательный разработчик и его работодатель.

Условно можно разделить тесты, которые выполняют программисты на протяжении всей разработки приложений, на три основных типа:

- 1) Модульные тесты (Unit tests)
- 2) Интеграционные тесты (Integration tests)
- 3) Сквозные тесты (End-to-end tests)

Каждый из перечисленных видов тестов используется на разных этапах разработки и имеет свои концептуальные особенности. Для применения этих тестов на практике используются различные стратегии тестирования. В данном случае, для достижения поставленных целей рассмотрены следующие стратегии:

- 1) Разработка на основе тестов (TDD или Test-driven development)
- 2) Разработка с учетом поведения (BDD или Behavior-driven development)

Суть стратегии TDD заключается в циклическом выполнении трёх стадий: написании тестового случая для проверки работы кода, изменении тестируемого кода, успешное выполнение тестового случая. Из данного подхода к разработке позже появился BDD. За основу взяты основные принципы TDD, однако здесь основной акцент делается на поведение отдельного программного модуля в различных сценариях, которые необходимы коду или клиенту, использующему данный модуль. Для того, чтобы оценить на практике работу таких подходов были взяты наиболее популярные ситуации, встречающиеся в разработке небольших приложений:

- 1) Тестирование бизнес-процессов
- 2) Тестирование в многопоточной среде
- 3) Тестирование взаимодействия с хранилищем данных

Для данной цели использовался язык JAVA, сторонние фреймворки JUnit, Mockito и системы сборки Maven. Достоинства следующие:

- код покрыт тестами и верифицирован;
- данные подходы заставляют разработчика заранее задумываться об архитектуре и качестве интерфейсов, иначе код будет сложно, а порой и невозможно, тестировать;
- в общем случае, затрачивается намного меньше времени на качественную разработку приложения;
- хорошие тесты могут легко заменить документацию, т.к. наглядно демонстрируют использование тестируемого кода;
- Недостатки у исследуемых подходов также присутствуют.
- высокий порог вхождения для разработчиков;
- ошибочный или некорректный тест приводит к написанию такого же ошибочного кода;
- много времени тратится при использовании интеграционных и сквозных тестов
- необходимо игнорировать слишком простые/сложные ситуации для тестирования;

Несмотря на то, что недостатки у подходов разработки через тестирование имеются, трудно переоценить преимущества, которые при этом получает разработчик и, как следствие, его работодатель.

Список использованных источников:

1. Tomek Kaczanowski Practical Unit Testing with Junit and Mockito
2. Савин Р. Г. Тестирование dotcom
3. Petar Tachiev, Felipe Leme, Vincent Massol, Gary Gregory Junit in action second edition
4. Exelixis Media P. C. Mockito Programming Cookbook

АНАЛИЗ ПАРАМЕТРОВ ПРОГНОЗИРОВАНИЯ ГЕНЕРАЦИИ ЭЛЕКТРОЭНЕРГИИ СОЛНЕЧНЫМИ ПАНЕЛЯМИ

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Степанов С.М.

Искра Н.А. – ст. преподаватель

В настоящее время проблема использования различных источников электрической энергии стоит особенно остро. Человечество использует невозобновляемые энергетические ресурсы, такие как нефть, газ, уголь, и объемы потребляемых ресурсов растут каждый год. Вследствие этого, количество невозобновляемых ресурсов сокращается, поэтому человечество начало использовать возобновляемые источники энергии, такие как солнечная энергия, энергия солнца, геотермальных источников и прочие.

Солнечная панель – полупроводниковое устройство, прямо преобразующее солнечную энергию в постоянный ток. Обычно солнечные панели объединяют в солнечные батареи. Схема солнечной панели представлена на рисунке 1[1]:

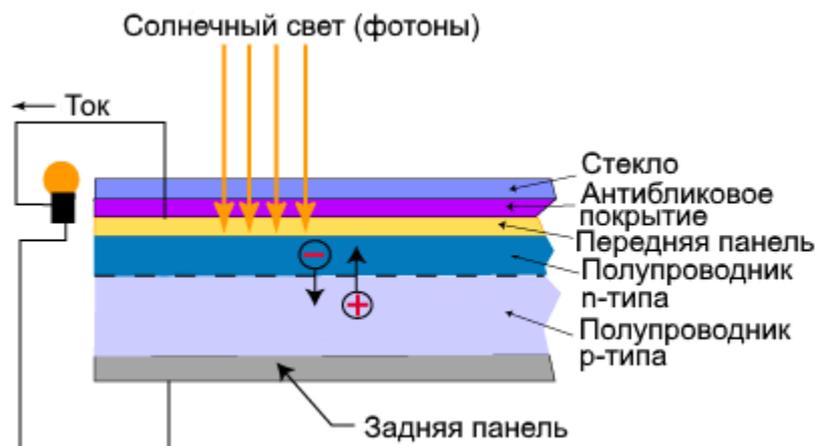


Рис. 1 – Схема солнечной панели

В настоящее время сложно сделать прогноз того, сколько электроэнергии сгенерируют солнечные батареи, т.к. количество электричества напрямую зависит от погоды и множества других факторов, таких как угол наклона панели к солнцу, длительность светового дня и др. Существуют интернет приложения, которые делают прогноз сгенерированной электроэнергии на следующие сутки, например, сайт бельгийской компании Elia[2].

Одним из возможных способов предсказания количества сгенерированной энергии является построение математической модели этого процесса. Другой способ – обучение нейросети на основании данных о генерации электроэнергии за предыдущие периоды.

Предлагается создать нейросеть двух различных типов: многослойный перцептрон Румельхарта и дерева принятия решений. Каждый из типов нейросети требуется обучить на одинаковых наборах данных. Наборы данных будут содержать один или несколько параметров, которые влияют на количество сгенерированной электроэнергии, в различных комбинациях:

- количество сгенерированной энергии;
- облачность;
- время рассвета и заката;
- интенсивность солнечного света;
- температура окружающей среды.

В дальнейшем, анализируя результаты нейросети о проценте верных прогнозов можно судить о том, какие наборы данных имеют наибольшее влияние на точность предсказания генерации электроэнергии солнечной батареей.

Список использованных источников:

1. LearnHowSolarPowerWork: [Электронный ресурс] // Как работают солнечные панели. URL <http://www.mysolarprojects.com/solar101.html>
2. "Solar-PVpowergenerationdata: [Электронный ресурс] // Данные о генерации энергии солнечными панелями. URL <http://www.elia.be/en/grid-data/power-generation/Solar-power-generation-data/Graph>

СИСТЕМА АНАЛИЗА КОНТЕКСТНОЙ ИНФОРМАЦИИ О РЫНКЕ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Садовников И.В.

Искра Н.А. – старший преподаватель

В настоящее время становится необходимо анализировать огромное количество данных, будь это данные о банковских транзакциях или данные поведения пользователя при выборе товаров и услуг. Классической схемой анализа данных с давних времен была та в которой человек сам анализировал поступающие данные, но с стремительным ростом количества данных этот способ стал изживать себя в силу ограниченности аналитических способностей человека. Поэтому появились новые технологии для обработки большого количества поступающих данных и позволяющие автоматизировать данный процесс.

График роста численности мобильных приложений с течением времени приведен на рисунке 1: