Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт космических и информационных технологий
Кафедра систем искусственного интеллекта

УТВЕРЖДАЮ
Заведующий кафедрой
_____ Г. М. Цибульский
подпись
«___» _____2018 г.

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

09.04.02.01 – «Информационно управляющие системы»

Разработка информационно-аналитической модели для блока управления электромотоцикла

Научный руководитель	_____ подпись, дата	проф., д-р физ.-мат. наук Б. С. Добронец
Выпускник	_____ подпись, дата	С. Д. Крицкий
Нормоконтролер	_____ подпись, дата	Б. С. Добронец

Красноярск 2018

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В современном мире электротранспорт завоевывает популярность в повседневном использовании, мировые производители ставят цели на следующее десятилетие выпустить полноценные серийные линейки электроавтомобилей и электромотоциклов. В городах проходят реформы о почти полной электрофикации транспорта, например, в Амстердаме.

Но в данное время электротранспорт без постоянного источника восполняемой энергии является по большей мере городским или локальным транспортом для преодоления небольших участков местности. Это обусловлено низким запасом хода электротранспорта, что не позволяет перемещаться на дальние расстояние и совершать, например, междугородние, патрулирующие или исследовательские поездки.

Исследования в области оптимизации потребления электроэнергии в электротранспорте являются актуальными, т.к. процент электротранспорта и его разновидность быстро растет и набирает популярность. Так же существует большое количество видов электродвигателей и электропотребление для них разное.

Решение практических задач по оптимизации расхода электроэнергии на электротранспорте является актуальной задачей и поможет улучшить продвижение в решении вопроса дальности хода электротранспорта.

Производители для улучшения характеристики запаса хода своего электротранспорта улучшают химические свойства батареи, работают над улучшением дизайна корпуса и рамы, увеличивают общий объем емкости батареи. В данной работе рассмотрен способ улучшения характеристики запаса хода рассматриваемого электромотоцикла при исходных данных ходовых характеристик и дизайна корпуса и рамы с помощью методов классификации и обработки данных.

1 Изучение темы

1.1 Актуальность рассматриваемой темы

В связи с ростом популярности использования электродвигателей в транспорте появляются новые потребности в улучшении и оптимизации характеристик данного вида транспорта.

Лидеры мирового рынка уже создают десятки моделей электроавтомобилей и электромотоциклов в различных вариациях и для разной аудитории, охват в социальных сетях растет по тегам «электромотор» или «электромобиль», что говорит о росте общей увлеченности людей данной темой. Ещё стоит принять во внимание показатель, что некоторые страны переходят или уже перешли на более чем 90% использования электротранспорта в гражданских целях.

В настоящее время исследования в области оптимизации потребления электроэнергии в электротранспорте являются актуальными, т.к. процент электротранспорта и его разновидность быстро растет и набирает популярность. Так же существует большое количество видов электродвигателей и электропотребление для них разное.

Решение практических задач по оптимизации расхода электроэнергии на электротранспорте является актуальной задачей и поможет улучшить продвижение в решении вопроса дальности хода электротранспорта.

Крупные производители электротранспорта решают проблему электропотребления с помощью улучшения химических свойств батарей, ускорения времени зарядки электротранспорта, улучшения особенностей конструкции корпуса электротранспорта [1].

1.2 Рассмотрение проблемы оптимизации электропотребления электромотоцикла

Оптимизация электропотребления является острой проблемой в современном мире. В данное время электротранспорт является по большей мере городским или локальным транспортом для преодоления небольших участков местности. Это обусловлено низким запасом хода и не позволяет перемещаться на дальние расстояние и совершать, например, межгородские, патрулирующие или исследовательские поездки.

Задача данной работы — увеличить запас хода рассматриваемого в данной работе электромотоцикла, при уже исходных характеристиках дизайна корпуса, архитектуры рамы и выбранного двигателя.

1.3 Цели и задачи работы оптимизации электропотребления для электромотоцикла

Целью работы является разработка информационно-аналитической модели для блока управления электромотиклом.

В задачи входит:

- Разработка алгоритма вычисления оптимизации расхода запаса батареи электромотоцикла.
- Разработка информационно-аналитической модели.
- Тестирование информационно-аналитической модели.
- Выводы по работе разработанной модели.

1.4 Объект исследования

Объектом исследования является электромотоцикл, построенный на базе Сибирского Федерального Университета [2]. Он имеет электродвигатель с напряжением 48В и мощностью 5 кВт, LiFePo4 батареи 48В на 60А/ч, между

ними работает синусный контроллер для управления моментом двигателя. При средней скорости в 70 км/ч, при полном заряде батареи запас хода данного ТС около 80 км. Задача — увеличить запас хода, при исходных характеристиках.

В данной работе смоделируем систему поведения электромотоцикла при различных условиях езды в условиях города. Ведь от характеристик дороги, уклона, фактора наличия «пробок» на дороге - все эти параметры влияют на конечный результат расхода батареи. Чтобы вычислить расход батареи требуется вычислить мощность для сохранения постоянной скорости электромотоцикла [3]:

Условие сохранения постоянной скорости ТС:

$$N_{\text{дв}} = N_{\text{в}} + N_{\text{д}} + N_{\text{п}} + N_{\text{к}}, \quad (1)$$

где $N_{\text{в}}$ – мощность, затрачиваемая при преодолении сил сопротивления ветра, кВт;

$\text{N}_{\text{д}}$ – мощность, затрачиваемая при преодолении сил сопротивления дороги, кВт;

$\text{N}_{\text{п}}$ – Мощность затрачиваемая при преодолении сил сопротивления подъема, кВт;

$\text{N}_{\text{к}}$ – Мощность затрачиваемая при преодолении сил сопротивления качению, кВт.

Мощность, затрачиваемая при преодолении сил сопротивления ветра:

$$N_{\text{в}} = \frac{k_{\text{в}} F_{\text{а}} (\upsilon \pm \upsilon_{\text{в}})^2 \upsilon}{1000}, \quad (2)$$

где $\text{k}_{\text{в}}$ – коэффициент сопротивления воздуха;

$\text{F}_{\text{а}}$ – лобовая площадь ТС, м^2;

υ –скорость ТС, м/с.

Мощность, затрачиваемая при преодолении сил сопротивления дороги:

$$N_{\text{д}} = \frac{\psi G \upsilon}{1000},$$

(3)

где ψ – коэффициент сопротивления дороги;

G – вес ТС, Н;

υ –скорость ТС, м/с.

Мощность, затрачиваемая при преодолении сил сопротивления подъема:

$$N_{\text{п}} = \frac{\upsilon G sin(\alpha)}{1000},$$

(4)

где υ –скорость ТС, м/с;

G – вес ТС, Н;

α – угол подъема в градусах.

Мощность, затрачиваемая при преодолении сил сопротивления качению:

$$N_{\text{к}} = \frac{\upsilon f_{\text{к}} G cos(\alpha)}{1000}, \quad (5)$$

где $f_{\text{к}} = \frac{115+\upsilon}{10000}$ – коэффициент сопротивления качению;

υ –скорость ТС, м/с;

α – угол подъема в градусах;

G – вес ТС, Н.

Тогда:

$$N_{\text{дв}} = \frac{k_{\text{в}} F_{\text{а}} (\upsilon \pm \upsilon_{\text{в}})^2 \upsilon}{10^3} + \frac{\psi G \upsilon}{10^3} + \frac{\upsilon G sin(\alpha)}{10^3} + \frac{\upsilon(115+\upsilon) G cos(\alpha)}{10^7} \quad (6)$$

Для расчета необходимой выдаваемой мощности электродвигателя при движении со скоростью 70 км/ч или 19,4 м/с используем представленную выше формулу и получим результат равный 1552 Вт, именно сколько чистой энергии нужно затратить на передвижение при исходных данных.

Производитель используемого электродвигателя предоставляет таблицу [3] тестирования динамики и её эффективность (Рисунок 1). Возьмем приближенную к нашему результату затрачиваемую энергию в 1531.11 Вт.

Items NO.	voltage V	current A	P. input W	P. factor PF	frequency Hz	torque mN.m	rotate rpm	P. output W	efficiency %
1	47.99	8.177	392.41	1.000	0.00	360.0	4389	165.45	42.2
2	47.98	8.538	409.70	1.000	0.00	242.5	4384	111.32	27.2
3	47.98	9.967	478.17	1.000	0.00	102.5	4369	46.89	9.8
4	47.95	13.222	633.99	1.000	0.00	577.5	4335	262.14	41.4
5	47.91	18.686	895.30	1.000	0.00	1412.5	4279	632.89	70.7
6	47.86	26.320	1259.60	1.000	0.00	2415.0	4204	1063.11	84.4
7	47.80	35.715	1707.06	1.000	0.00	3552.5	4116	1531.11	89.7
8	47.72	46.523	2219.96	1.000	0.00	4812.5	4021	2026.29	91.3
9	47.63	58.475	2785.48	1.000	0.00	6182.5	3923	2539.68	91.2

Рисунок 1 - Тест динамики двигателя

По таблице видно, что требуется учитывать КПД двигателя, следовательно, нужно потратить не 1531 Вт чистыми, а 1707 Вт энергии. В результате, для движения по ровной дороге со скоростью 70 км/ч требуется 1707 Вт мощности системы.

К примеру, необходимо проехать 100 км с данными характеристиками. При скорости 70км/ч 100 км преодолевается за 1,4 часа.

1,4ч * 1707Вт = 2389 В*А*ч.

Поделим и получим приблизительное значение необходимой емкости батареи:

3329 В*А*ч / 48В = 49Ач

Но известно, что в рассматриваемом электромотоцикле стоят батареи на 60Ач, тогда при 100/69*60 получаем примерно 122 км дальности хода по асфальтированной прямой дороге без наклонов.

Чтобы повлиять на электропотребление данного электромотоцикла доступен регулируемый параметр — напряжение ручки газа, которая используется как потенциометр для регулировки крутящего момента электродвигателя, интервал действия ручки газа от 1 до 5В.

То есть если выкрутить ручку газа на половину, это даст 2,5В напряжения на контроллер, что даст электродвигателю половину возможного крутящего момента (14 Нм), а чем больше крутящий момент, тем больше расход батареи. В условиях движения в черте города встречаются внешние факторы, влияющие на потребление батареи: это подъемы, спуски и автомобильные пробки. Погодные условия в виде низкой или высокой температуры в расчет брать не будем в рамках данной задачи.

1.5 Обзор научных статей по рассматриваемой теме

1) Повышение эффективности работы электроцентробежных насосов на месторождениях с трудноизвлекаемыми запасами нефти. Авторы: Караичев К. Н., Зырин В. О. Расположение публикации: Национальный минерально-сырьевой университет «Горный», журнал EUROPEAN SCIENCE 2016 г. [5]

В статье рассмотрена проблема увеличения эффективности электродвигателей насосов для механической добычи нефти на месторождениях. Путем создания новой системы управления электродвигателем, которая позволит снизить энергопотребление. Также улучшения нефтедобычи на месторождениях с трудноизвлекаемыми запасами нефти можно достичь за счет оптимизации системы «интеллектуального месторождения» - системы автоматического управления операциями по добыче нефти и газа, в рамках которой создаются интегральная модель месторождения

и модель управления добычей. Данная система позволяет контролировать и прогнозировать нефтедобычу.

Авторы пришли к выводу, что одним из способов повышения эффективности работы насосов на месторождениях с трудноизвлекаемыми запасами нефти является разработка энергоэффективной системы управления насосными установками - системы с непрерывным контролем, адаптивным управлением, с использованием современных и быстродействующих управляющих устройств – таких как контроллеры, ПЛИС – программируемые логические интеллектуальные системы.

2) Энергоэффективное управление двигателем последовательного возбуждения. Авторы: Мин Т. А., Суздорф В. И. Расположение публикации: журнал Вестник Кузбасского Государственного Технического университета, 2017 г. [6]

Статья посвящена научной проблематике, связанной с оптимизацией энергетики, алгоритмизации управления, анализа динамических свойств и управления электроприводом с двигателем последовательного возбуждения в переходных процессах для поиска оптимальных управлений с точки зрения потерь энергии. Электродвигатели последовательного возбуждения широко применяются в различных сферах от тягового электропривода до бытовых приборов и ручного электроинструмента, результаты исследования, приведенные в статье, констатируют наличие научной проблематики, связанной с оптимизацией энергетики подобных систем. Из целого комплекса проблем оптимизационной задачи в статье поставлен вопрос энергоэффективности управления. При анализе электромагнитных процессов и функциональном синтезе управлений использованы принцип динамического программирования и спектральный метод В.В. Солодовникова, а также моделирование в среде Matlab Simulink.

3) Практический алгоритм численного поиска для минимизации мощности потерь асинхронного электродвигателя. Авторы: Борисевич А. В. Размещение статьи: журнал “Науковеденье”, 2014 г. [7]

Цель данной статьи - разработка метода прямого численного поиска минимума энергопотребления электродвигателя, который бы обладал достаточной концептуальной и вычислительной простотой для реализации на основе микропроцессоров общего назначения. Описанное в статье моделирование системы оптимизации создано в среде MATHLAB/Simulink, там же проведены эксперименты с реальным электродвигателем. Результаты показали, что метод работоспособен и устойчив к шумам измерения.

4) Анализ методов повышения энергоэффективности электродвигателей в машиностроении. Авторы: Золотых С.Ф., Рожков С.В., Лобанова С.В. Расположение статьи: Журнал Известия Тульского государственного университета. Технические науки. 2013 г. [8]

В данной статье в результате литературно патентного анализа доступных в настоящее время технологий повышения энергоэффективности электродвигателей, преимущественно применяемых в машиностроении, приведены наиболее хорошо зарекомендовавшие себя и экономически эффективные методы повышения энергоэффективности электроприводов. Каждый из выбранных методов аргументирован.

1.6 Сравнительный анализ выбранных публикаций по теме исследования

Для анализа выбранных статей воспользуемся методом сравнения. Составим таблицу (Таблица 1) для наглядного сравнения выбранных статей, критерии будут такими: год публикации, использованные методы, основная проблема работы и задачи, решаемые автором.

Таблица 1 - Сравнительная таблица по выбранным публикациям

Год	Название статьи	Авторы	Использованный метод	Задачи, решаемые автором	Основная проблема
2016	Повышение эффективности работы электроцентробежных насосов на месторождениях с трудноизвлекаемыми запасами нефти	Караичев К. Н., Зырин В. О. .	Предложение создать систему для контролирования и прогнозирования нефтедобычи	Задача увеличения эффективности электродвигателей насосов нефтедобычи	проблема увеличения эффективности электродвигателей насосов для механической добычи нефти на месторождениях
2017	Энергоэффективное управление двигателем последовательного возбуждения	Мин Т. А., Суздорф В. И	Метод В.В.Солодовникова, а также моделирование в среде Matlab Simulink.	Задача анализа динамических свойств и управления электроприводом с двигателем последовательного возбуждения в переходных процессах для поиска оптимальных управлений с точки зрения потерь энергии	Проблема энергоэффективности управления
2014	Практический алгоритм численного поиска для минимизации мощности потерь асинхронного электродвигателя	Борисевич А. В.	Моделирование системы оптимизации в среде MATHLAB/Simulink	Разработка метода прямого численного поиска минимума энергопотребления электродвигателя	Проблема потери мощности электродвигателя при работе
2013	Анализ методов повышения энергоэффективности электродвигателей в машиностроении	Золотых С.Ф., Рожков С.В., Лобанова С.В.	Увеличение кпд электродвигателя путем подсчета мощности.	Увеличение эффективности электродвигателя	Дорогая эксплуатация и малая эффективность электродвигателей

По получившейся таблице можно сделать следующие выводы:

1) Самая ранняя статья из выбранных публикаций была издана в 2013 году, что говорит об актуальности данной темы в настоящее время;

2) В рассматриваемых статьях авторы описывают разные проблемы, связанные с увеличением эффективности работы электродвигателя путем оптимизации работы с ним. Что говорит о том, что использование электродвигателей обширно и есть общая проблема улучшения их производительности.

Дополнительно были рассмотрены такие статьи как «Анализ влияния параметров линейного импульсного электродвигателя на эффективность его работы», авторы Милых В.И., Ткаченко С.В.[9]

«Разработка и исследование энегосберегающей системы управления регулируемыми электродвигателями насосов кнс», авторы Лысова О.А., Портнягин А.Л., Кочнев П.В.[10].

«Система управления электродвигателем гребной электрической установки», авторы Калмыков А.Н, Галушин С.Я., Дмитриев Б.Ф.[11].

«Предложения по улучшению системы автоматического управления током тяговых электродвигателей электровоза для обеспечения его работоспособности», авторы Савоськин А. Н., Телегин М. В., Плис В. И., Жирков А. И.[12]

Система прямого управления моментом и потокосцеплением ротора асинхронного электродвигателя, авторы Овсянников Е.М., Нгуен К.Т. [13].

«Имитационная модель системы «преобразователь частоты - асинхронный электродвигатель» насосов системы водоснабжения с адаптивным алгоритмом управления», авторы Таранов Д.М., Каун О.Ю., Гетманенко В.М., Лебедев К.Н., Литвинов В.Н., Старовойт М.Ю.[14]

«Об исследовании динамики систем управления электродвигателем», авторы Яночкина О. О. [15].

«Моделирование систем автоматического управления электродвигателя робота-конвертоплана», авторы Москера М. М. [16].

«Проектирование системы автоматического управления электродвигателем постоянного тока», авторы Максаков С.А. [17].

«Параметры режимов асинхронных электродвигателей в системе умного управления», авторы Минакова Т.Е. [18].

«Разработка и реализация на плис энергоэффективных способов импульсного управления системами "усилитель мощности - электродвигатель" на основе методов автоматизированного проектирования», авторы Кривилев А.В. [19].

«Цифровые системы управления асинхронным электродвигателем», авторы Захаров В.С. [20]

Исходя из затрагиваемых в статьях проблемах можно сделать вывод, что основные проблемы совпадают с проблемой, выделенной в данной научно-исследовательской работе, т.е. оптимизация работы электродвигателя и повышение его эффективности. Исследования в области электродвигателей проходят во многих областях — в нефтедобыче, в авиации, в аграрной области, в судоходстве, промышленные объекты, транспорт. Большинство сходится в том, что в современном мире для поддержания и управлением электродвигателем требуется создавать системы управления для повышения эффективности и оптимизации работы электродвигателя. Предлагается также создавать системы с возможностью прогнозирования и самообучением.

Выбор данных публикаций был обоснован наличием теоретических сведений, и методов с полными решениями необходимых задач, что необходимо на начальном этапе исследования темы. Так же проблемы, и предметы исследования, выявленные авторами в выбранных публикациях, совпадают с проблемами, исследуемыми в рамках данной работы.

Результат проведенного сравнительного анализа статей свидетельствует о том, что для решения проблемы научного исследования разрабатываются

специальные методы, из которых выбираются самые эффективные для оптимизации работы электродвигателя и повышения его эффективности. Для этого проводятся модификации существующих методов под конкретные процессы и условия.

2 Методы классификации

2.1 Методы решения задач классификаций

Задача классификации — формализованная задача, в которой имеется множество объектов (ситуаций), разделённых некоторым образом на классы. Задано конечное множество объектов, для которых известно, к каким классам они относятся. Это множество называется выборкой. Классовая принадлежность остальных объектов неизвестна. Требуется построить алгоритм, способный классифицировать (см. ниже) произвольный объект из исходного множества.

Классифицировать объект — значит, указать номер (или наименование) класса, к которому относится данный объект. Классификация объекта — номер или наименование класса, выдаваемый алгоритмом классификации в результате его применения к данному конкретному объекту.

В математической статистике задачи классификации называются также задачами дискриминантного анализа. В машинном обучении задача классификации решается, в частности, с помощью методов искусственных нейронных сетей при постановке эксперимента в виде обучения с учителем [21].

2.2 Обзор методов задач классификации

2.2.1 Решающее дерево

Решающие деревья воспроизводят логические схемы, позволяющие получить окончательное решение о классификации объекта с помощью ответов на иерархически организованную систему вопросов. Причём вопрос, задаваемый на последующем иерархическом уровне, зависит от ответа, полученного на предыдущем уровне. Подобные логические модели издавна используются в ботанике, зоологии, минералогии, медицине и других областях. Пример, решающего дерева, позволяющая грубо оценить стоимость квадратного метра жилья в предполагаемом городе приведена на Рисунке 2. Схеме принятия решений, изображённой на Рисунке 2, соответствует связный ориентированный ациклический граф – ориентированное дерево. Дерево включает в себя корневую вершину, инцидентную только выходящим рёбрами, внутренние вершины, инцидентную одному входящему ребру и нескольким выходящим, и листья – концевые вершины, инцидентные только одному входящему ребру

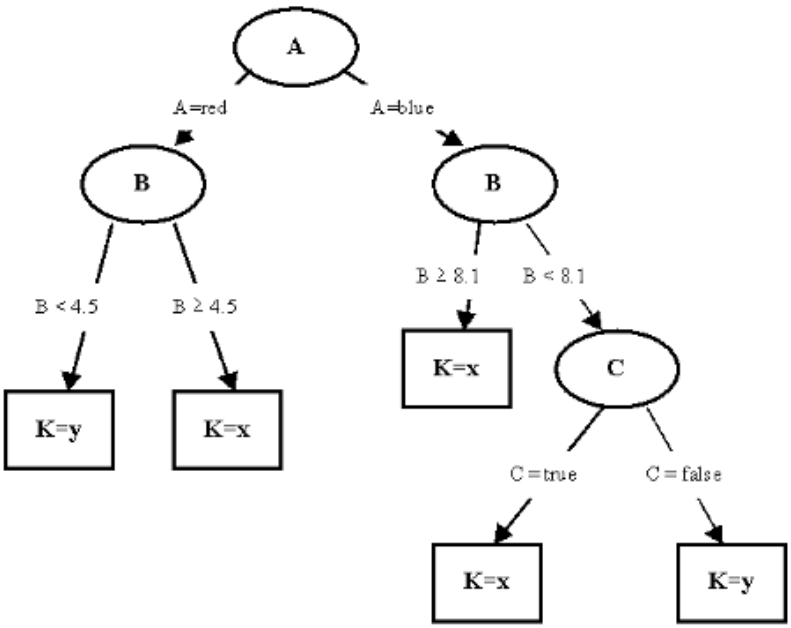

Рисунок 2 - Схема решающего дерева

Каждой из вершин дерева за исключением листьев соответствует некоторый вопрос, подразумевающий несколько вариантов ответов, соответствующих выходящим рёбрам. В зависимости от выбранного варианта ответа осуществляется переход к вершине следующего уровня. Концевым вершинам поставлены в соответствие метки, указывающие на отнесение распознаваемого объекта к одному из классов. Решающее дерево называется бинарным, если каждая внутренняя или корневая вершина инцидентна только двум выходящим рёбрам. Бинарные деревья удобно использовать в моделях машинного обучения.

2.2.2 Метод ближайших соседей

Метод решения задачи классификации, который относит объекты к классу, которому принадлежит большинство из k его ближайших соседей в многомерном пространстве признаков. Это один из простейших алгоритмов обучения классификационных моделей. Число k – это количество соседних объектов в пространстве признаков, которое сравнивается с классифицируемым объектом. Иными словами, если k=10, то каждый объект сравнивается с 10-ю соседями. Метод широко применяется в технологиях Data Mining для решения задач классификации.

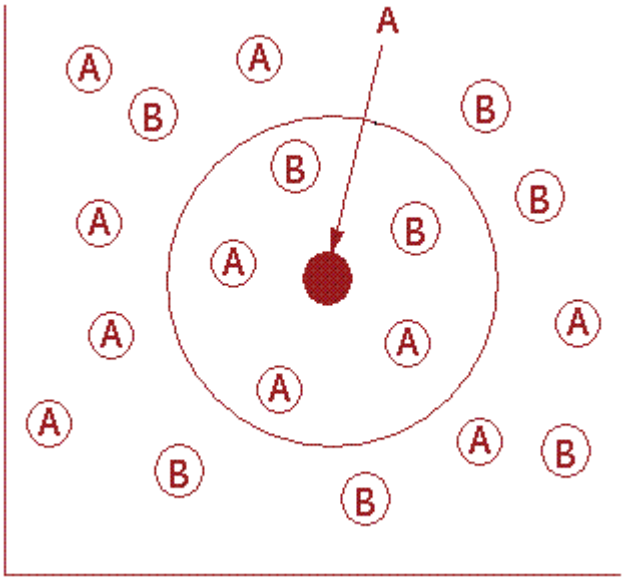

Рисунок 3 - Схема метода ближайших соседей

В процессе обучения алгоритм просто запоминает все векторы признаков и соответствующие им метки классов. При работе с реальными данными, т.е. наблюдениями, метки класса которых неизвестны, вычисляется расстояние между вектором нового наблюдения и ранее запомненными. Затем выбирается k ближайших к нему векторов, и новый объект относится к классу, которому принадлежит большинство из них.

Выбор параметра k противоречив. С одной стороны, увеличение его значения повышает достоверность классификации, но при этом границы между классами становятся менее четкими. На практике хорошие результаты дают эвристические методы выбора параметра k, например, перекрестная проверка.

Несмотря на свою относительную алгоритмическую простоту метод показывает хорошие результаты. Главным его недостатком является высокая вычислительная трудоемкость, которая увеличивается квадратично с ростом числа записей в наборе данных.

2.2.3 Метод окна Парзена

Метод байесовской классификации, основанный на непараметрическом восстановлении плотности по имеющейся выборке. После ввода метрики, метод парзеновского окна можно использовать, не опираясь на вероятностную природу данных.

В основе подхода лежит идея о том, что плотность выше в тех точках, рядом с которыми находится большое количество объектов выборки.

Если мощность множества элементарных исходов много меньше размера выборки, то в качестве восстановленной по выборке плотности мы вполне можем взять и гистограмму значений выборки.

В противном случае (например, непрерывном) данный подход не применим, так как плотность концентрируется вблизи обучающих объектов, и функция распределения претерпевает резкие скачки. Приходится использовать восстановление методом Парзена-Розенблатта.

2.2.4 Непараметрическая регрессия. Формула Надарая-Ватсона

В отличие от параметрических подходов, использует модель, которая не описывается конечным числом параметров.

Цель регрессионного анализа состоит в осуществлении разумной аппроксимации неизвестной функции отклика $Y(X)$ по известным точкам . $(X_i, Y_i)_{i=1}^{m}$. В случае малых ошибок наблюдения становится возможным сконцентрировать внимание на важных деталях средней зависимости Y от X при ее интерпретации.

Процедура аппроксимации обычно называется сглаживанием. По существу эта аппроксимация функции отклика Y может быть выполнена двумя способами. Довольно часто используется параметрический подход, заключающийся в предположении, что функция отклика Y имеет некоторую предписанную функциональную форму, например, это прямая линия с неизвестными свободным членом и наклоном. Альтернативой этому может

служить попытка оценить Y непараметрическим образом, без указания конкретного ее вида. Первый подход к анализу регрессионной зависимости называется параметрическим, поскольку предполагается, что вид функции полностью описывается конечным набором параметров. Типичный пример параметрической модели представляет собой полиномиальное уравнение регрессии, когда параметрами являются коэффициенты при неизвестных.

Однако при параметрическом подходе молчаливо предполагается, что кривая может быть представлена в терминах параметрической модели, или, по крайней мере, имеется уверенность в том, что ошибка аппроксимации для наилучшего параметрического приближения пренебрежимо мала. Наоборот, в непараметрической модели регрессионной зависимости не производится проектирования данных в "прокрустово ложе" фиксированной параметризации.

Предварительное задание параметрической модели может оказаться слишком ограничительным или чересчур малой размерности для аппроксимации непредвиденных характеристик, в то время как непараметрическое сглаживание предоставляет гибкие средства анализа неизвестных регрессионных зависимостей.

Непараметрический подход приводит, таким образом, к гибкому функциональному виду кривой регрессии.

2.2.5 Метод опорных векторов

Является одной из наиболее популярных методологий обучения по прецедентам, предложенной В. Н. Вапником и известной в англоязычной литературе под названием SVM (Support Vector Machine).

Оптимальная разделяющая гиперплоскость. Понятие зазора между классами (margin). Случай линейной разделимости. Задача квадратичного программирования. Опорные векторы. Случай отсутствия линейной разделимости. Функции ядра (kernel functions), спрямляющее пространство, теорема Мерсера. Способы построения ядер. Примеры ядер. Сопоставление

SVM и нейронной RBF-сети. Обучение SVM методом активных ограничений. SVM-регрессия.

В случае метода опорных векторов точка в пространстве рассматривается как вектор размерности p. Мы хотим узнать, сможем ли мы разделить данные точки гиперплоскостью размерности (p-1). Ее назовем линией классификатора. Данные могут быть классифицированы с помощью различных гиперплоскостей. Лучшая гиперплоскость - гиперплоскость, при построении которой разделение и разница между 2 классами максимально.

Двумерные объекты – данные на плоскости. Рассмотрим данные на плоскости, гиперплоскость в данном случае – это прямая. Проведем любую прямую, она разделит точки на 2 множества. Выберем прямую, максимально далеко проходящую от точек. Расстояние от нее до ближайшей точки с каждой стороны максимальна. Если такая прямая существует, то ее называют гиперплоскость максимальной разницы, и линейный классификатор, ее определяющий, соответственно классификатор максимальной разности или персептрон оптимальной стабильности (устойчивости).

Опорные вектора – это точки, для которых расстояние до гиперплоскости. Они являются эффективными элементами на обучающей выборке.

2.2.6 Метод случайный лес

Алгоритм машинного обучения, предложенный Лео Брейманом и Адель Катлер, заключающийся в использовании комитета (ансамбля) решающих деревьев. Алгоритм сочетает в себе две основные идеи: метод бэггинга Бреймана, и метод случайных подпространств, предложенный Tin Kam Ho. Алгоритм применяется для задач классификации, регрессии и кластеризации. Основная идея заключается в использовании большого ансамбля решающих деревьев, каждое из которых само по себе даёт очень невысокое качество классификации, но хороший результат получается за счёт больших чисел.

Рассмотрены самые популярные методы классификации данных. У каждого метода есть как свои плюсы, так и минусы. Каждый метод по - своему подходит к различным задачам. В рассматриваемой задаче имеется три переменные в формуле расчета потребляемой энергии батареи, влияющие на показатель эффективности электродвигателя.

2.3 Python для задач анализа данных

Для работы с числовыми данными будет использоваться язык программирования Python, так как он уже является на сегодняшний день фактическим стандартом для анализа данных и математических операций [22].

Предполагается использовать следующие библиотеки:

- NumPy - это библиотека языка Python, добавляющая поддержку больших многомерных массивов и матриц, вместе с большой библиотекой высокоуровневых (и очень быстрых) математических функций для операций с этими массивами.

- SciPy - библиотека предназначенная для выполнения научных и инженерных расчётов.

- Matplotlib - библиотека на языке программирования Python для визуализации данных двумерной (2D) графикой (3D графика также поддерживается). Получаемые изображения могут быть использованы в качестве иллюстраций в публикациях.

- Pandas - библиотека для анализа данных.

- Scikit-learn - это библиотека для реализации машинного обучения.

Для первоначальной разработки программы предлагается использовать дистрибутив Python под названием Anaconda. Преимущества Anaconda в том, что она содержит в себе уже более 200 библиотек, в том числе вышеперечисленные [23]. Запускается с помощью браузера, что делает возможным запуск с любой операционной системы (которая поддерживает работу с браузером).

2.4 Генерация тестовых данных

Так как рассматривается три переменных для анализа, то можно сгенерировать их средствами Python и с помощью numpy. Первоначально сгенерируем данные по углу наклона дороги. Графически продемонстрирует данные в виде гистограммы Matplotlib.

Листинг:

```
import numpy as np
import matplotlib.mlab as mlab
import matplotlib.pyplot as plt
mu = 0.4  # среднее значение угла наклона
sigma = 0.15  # стандартное отклонение значение
x = mu + sigma * np.random.randn(1000)
num_bins = 80 # количество разделений данных на бункеры
#Отрисовка гистограммы
n, bins, patches = plt.hist(x, num_bins, normed=1, facecolor='orange', alpha=0.5)
y = mlab.normpdf(bins, mu, sigma)
plt.plot(bins, y, 'r--')
plt.xlabel('Угол')
plt.ylabel('Вероятность')
plt.title(r'Данные по углу наклона дороги')
plt.subplots_adjust(left=0.15)
plt.show()
```

Результат по углу наклона дороги в виде гистограммы представлен на Рисунке 4.

Рисунок 4 - Гистограмма угла наклона дороги

Те же действия необходимо проделать для второй переменной - коэффициента силы трения поверхности дороги. Среднее значение в городе данного коэффициента 0,018 (асфальт).

mu = 0.018 # среднее значение коэффициента трения

sigma = 0.009 # стандартное отклонение значение

Результат в виде гистограммы представлен на Рисунке 5:

Рисунок 5 - Гистограмма коэффициента трения дороги

Генерация тем же методом данных по скорости транспорта. Скорость в итоговом варианте должна возводиться в куб по условиям формулы.

mu = 60 # среднее значение скорости

sigma = 10 # стандартное отклонение значение

Результат выполнения генерации данных в виде гистограммы:

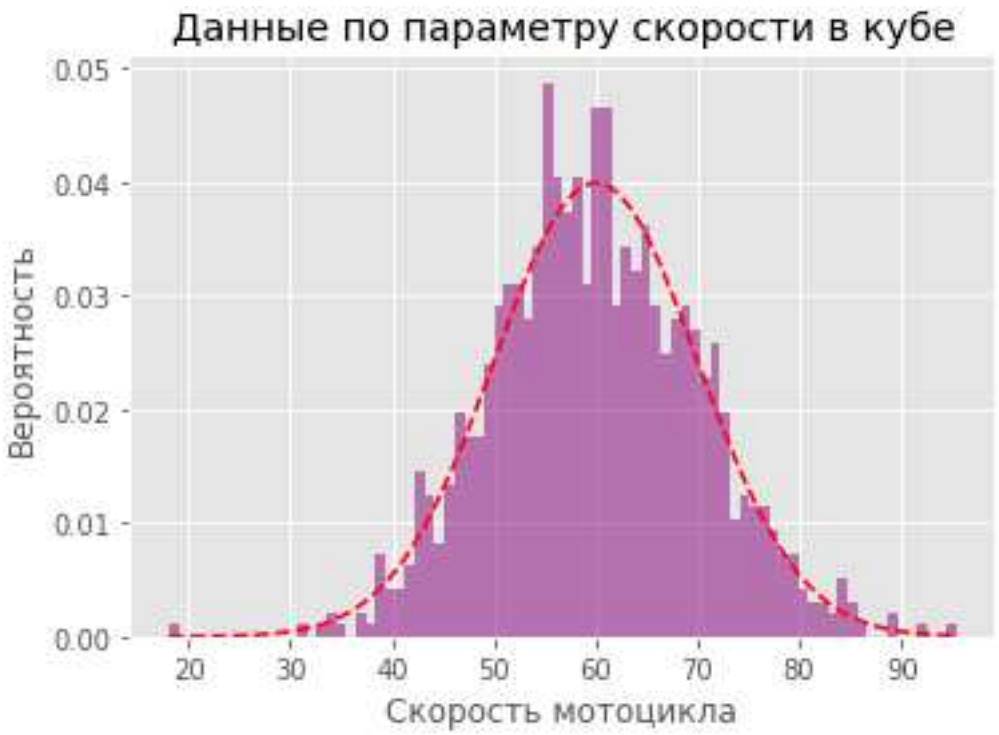

Рисунок 6 - Гистограмма скорости мотоцикла

Генерируемые файлы будем записывать в list.

Листинг:

```
mu = 60 # среднее значение скорости
sigma = 10  # стандартное отклонение значение
speed_array = []
x = mu + sigma * np.random.randn(10)
for (i, item) in enumerate(x):
speed_array.append(item)
for (i, item) in enumerate(speed_array):
print (i+1, item)
```

Для ускорения операций уменьшаем количество итераций до 10-ти. Результат хранимых данных в списке представлен на Рисунке 7:

```
for (i, item) in enumerate(speed_array):
    print (i+1, item)
```

```
1 68.81875103322184
2 82.40186902713594
3 61.842809012597066
4 73.39897393013959
5 64.54675026805667
6 58.287942562912264
7 64.91670648591874
8 65.49454842838107
9 69.64688448528864
10 69.88989488998462
```

Рисунок 7 - Результат сохранения данных (скорость)

2.5 Применение стохастических данных

В работе рассматривается формула для расчета необходимой выдаваемой мощности электродвигателя при движении со скоростью 70 км/ч или 19,4 м/с используется формула 1. То есть, если составить итоговую формулу, то получится такой вид, листинг:

```
cx = 0.342
S = 0.5
g = 9.8
m = 230
pb = 1.225
for i in range(1,10):
        speed = float("{0:.1f}".format(speed_array[i]))
        fric = float("{0:.1f}".format(friction_array[i]))
        angle = float("{0:.1f}".format(angle_array[i]))
W=g*fric*m*(speed*1000/3600)*np.cos(angle)+0.5*cx*S*pb*((speed*1000/3600)**3)*g*m*np.sin(angle)*(speed*1000/3600)
```

```
print((int(float("{0:.1f}".format(W/1000)))))
```

Результат выполнения на десять итераций (Вт):

```
19141
0
11025
4417
2165
3800
10149
4765
1711
```

Как видно из списка в выборке могут попадаться пустые значения, значит потребуется очистка данных.

Выше описанный, код в общей структуре программы будет иметь вид отдельной функции для расчета требующейся мощности на каждую итерацию. Для определения класса эффективности будет использоваться метод случайного леса (random forest).

В элемент «леса» входит:

- функция мощности (Вт);
- функция расчета эффективности;
- определение класса (эффективный/не эффективный);

2.6 Обзор комплектующих для модуля управления

На текущий момент в блоке управления рассматриваемого электромотоцикла установлена платформа Arduino на базе микроконтроллера ATmega2560, предназначенная для управления периферии транспорта и для синхронизации с android устройством.

Для дальнейшего развития блока управления с добавлением информационно-аналитического модуля требуется заменить Arduino, по

причине нехватки мощности и не подходящей среды разработки под выбранный язык программирования Python.

Для использования Python на микроконтроллерах подходят платы с ARM (Advanced RISC Machine) архитектурой.

RISC – это архитектура процессора, в котором быстродействие увеличивается за счёт упрощения инструкций, чтобы их декодирование было более простым, а время выполнения — меньшим.

Из доступных микроконтроллеров на рынке есть платы линейки stm32 и платы Raspberry Pi.

На Raspberry Pi присутствует возможность установки Unix-подобных систем, в том числе созданную специально ОС Raspbian для упрощенного управления модулями платы, присутствует поддержка Python 2, 3 версий.

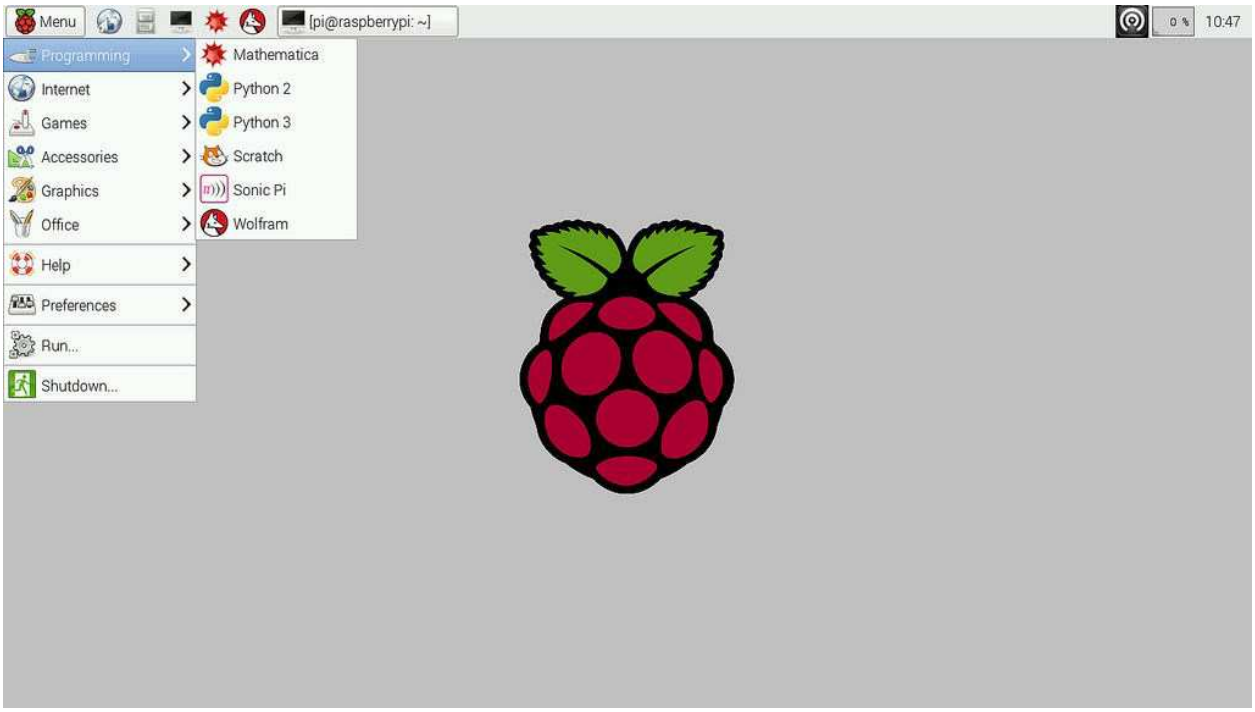

Рисунок 8 - Рабочий стол OS Raspbian

Так как Raspberry в общей схеме модуля находится внутри корпуса электромотоцикла и соединена в цепь с датчиками и питанием, то ее отладка с извлечением из общей цепи очень затруднительна, поэтому используется метод удаленного доступа к рабочему столу компьютера по сети VNC, что позволяет

отлаживать и обновлять программную часть модуля на любом расстоянии, без демонтажа общей цепи, при условии подключения Raspberry Pi к сети.

2.7 Модуль управления

Информационно-аналитический модуль включает в себя:

- Плату с микропроцессором - Raspberry Pi 3B+.
- Набор датчиков – датчик холла, гироскоп.
- Программный код на Python.

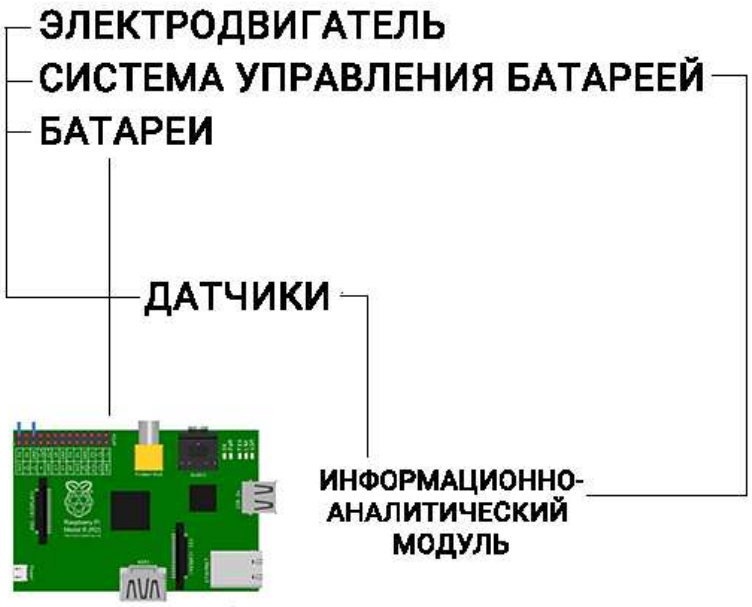

Рисунок 9 - Общая схема электромотоцикла

Из общей схемы для нашей работы понадобятся датчики, как источник входящих данных и информационно-аналитический модуль. Получая результат работы модуля, напрямую влиять на электромотоцикл не рекомендуется, так как создавая влияние на электропотребление прямым воздействием, происходит влияние на текущую скорость, что является не безопасным действием.

Вывод по главе 2

В данной главе рассмотрены методы классификации данных и выбран наиболее подходящий метод для данной разработки. Выбран язык программирования Python и вычислительный микрокомпьютер Raspberry PI для реализации работы информационно-аналитического модуля. Составлен список используемых библиотек языка Python для использования при разработке. Проведены первые работы по генерации тестовых данных на дистрибутиве Python Anaconda.

3 Разработка программы

3.1 Структура программы

Обозначим компоненты информационно-аналитического модуля, которые потребуются для его работы:

а) Генератор значений тестовых данных.

б) Обработка данных.

в) Математический блок моделирования.

г) Математический блок ассистента.

д) Хранение данных.

е) Прием входящих данных (GPIO).

ж) Графический блок (гистограммы, таблицы, графики).

Используемые библиотеки Python:

1) NumPy - это библиотека языка Python, добавляющая поддержку больших многомерных массивов и матриц, вместе с большой библиотекой высокоуровневых (и очень быстрых) математических функций для операций с этими массивами.

2) SciPy - библиотека предназначенная для выполнения научных и инженерных расчётов.

3) Matplotlib - библиотека на языке программирования Python для визуализации данных двумерной (2D) графикой (3D графика также поддерживается). Получаемые изображения могут быть использованы в качестве иллюстраций в публикациях.

4) Pandas - библиотека для анализа данных.

5) Scikit-learn - библиотека для реализации машинного обучения.

6) RPi.GPIO - библиотека созданная для управления пинами на плате Raspberry Pi.

3.2 Описание блоков программы

Для разработки используем IDE (интегрированная среда разработки) Pycharm Community.

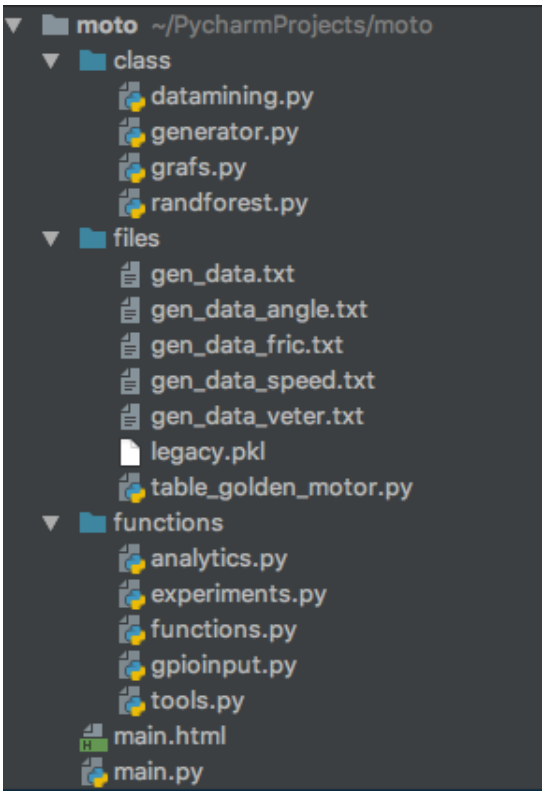

Рисунок 10 - Структура программы

Для написания программы используем объектно-ориентированное программирование (ООП). ООП используется для повышения уровня

читабельности и понимания кода, для разбиения на блоки и возможностью использовать уровень абстракции.

ООП структура программы позволит разделить проект на часть моделирования данных и на часть практического применения на электромотоцикле, без потери качества и времени работы над проектом.

Описание классов:

а) Generator.py – файл класса с входными значениями переменных для генерации тестовых данных.

б) Datamining.py – файл класса для обработки и классификации данных.

в) Grafs.py – файл класса для отображения графиков, основная библиотека Matplotlib.

г) Randforest – файл класса для работы с методом машинного обучения Случайный лес с применением библиотеки Scikit-learn.

Файлы:

1) Файл main.py – главный файл инициализации программы.

2) Файл functions.py – служит для содержания используемых математических функции в программе.

3) Файл tools.py – служит для содержания вспомогательных функций программы.

4) Analytics.py – файл для графического отображения полученных данных с датчиков.

5) Файл table_golden_motor.py – файл, содержащий в словаре таблицу с характеристиками двигателя от поставщика.

3.2.1 Генератор значений

Для генерации тестовых данных написан следующий класс программы Generator.py:

```
class Generator:
    m_speed = 0  # среднее значение скорости
```

```
    sigma_speed = 0  # стандартное отклонение значение
    mu_angle = 0  # среднее значение угла наклона
    sigma_angle = 0  # стандартное отклонение значение
    mu_trenie = 0  # среднее значение коэффициента трения
    sigma_trenie = 0  # стандартное отклонение значение
    count_step = 1000
    count_range = 1000
    def __init__(self, mu_speed, sigma_speed, mu_angle, sigma_angle, mu_trenie, sigma_trenie,
mu_veter, sigma_veter):
        self.mu_speed = mu_speed
        self.mu_angle = mu_angle
        self.mu_trenie = mu_trenie
        self.mu_veter = mu_veter
        self.sigma_speed = sigma_speed
        self.sigma_angle = sigma_angle
        self.sigma_trenie = sigma_trenie
        self.sigma_veter = sigma_veter
        self._gen_arrays()
    def _gen_arrays(self):
        speed_array = []
        friction_array = []
        angle_array = []
        veter_array = []
        speed = self.mu_speed + self.sigma_speed * np.random.randn(self.count_step)
        angle = self.mu_angle + self.sigma_angle * np.random.randn(self.count_step)
        friction = self.mu_trenie + self.sigma_trenie * np.random.randn(self.count_step)
        veter = self.mu_veter + self.sigma_veter * np.random.randn(self.count_step)
        tools.write_to_file('files/gen_data_speed.txt', speed)
        tools.write_to_file('files/gen_data_angle.txt', angle)
        tools.write_to_file('files/gen_data_fric.txt', friction)
        tools.write_to_file('files/gen_data_veter.txt', veter)

        for (i, item) in enumerate(speed):
            speed_array.append(item)
```

```
        for (i, item) in enumerate(friction):
            friction_array.append(item)
        for (i, item) in enumerate(angle):
            angle_array.append(item)
        for (i, item) in enumerate(veter):
            veter_array.append(item)
        self._math_array(speed_array, friction_array, angle_array, veter_array)
    def _math_array(self, speed_array, friction_array, angle_array, veter_array):
        file = open('files/gen_data.txt', 'w')
        for i in range(1, self.count_range):
            speed = float("{0:.4f}".format(speed_array[i]))
            fric = float("{0:.4f}".format(friction_array[i]))
            angle = float("{0:.4f}".format(angle_array[i]))
            veter = float("{0:.4f}".format(veter_array[i]))
            speed = speed * 1000 / 3600
            N = functions.save_lumbda_speed(speed, fric, angle, veter)
            file.write(str((int(float("{0:.4f}".format(N * 1000)))))+'\n')
        file.close()
```

В данном классе мы генерируем данные для параметров скорости, угла наклона дороги, коэффициента трения дороги и ветра с указанным разбросом значений.

Экземпляр класса создается в главном файле main.py: generator.Generator(50, 30, 4, 2, 0.015, 0.009, 4, 10).

Входные данные взяты усредненные и приближенные к реальности. Для этого были изучены материалы по видам наклонов дорог от министерства транспорта, взяты известные значения трения поверхности дорог и более вероятные повседневные значения скорости ветра. Все полученные значения параметров записываются в отдельные файлы с расширением txt. Для дальнейших расчетов вызывается функция сохранения скорости электромотоцикла из блока математических функций function.py.

Все результаты вычисления формулы записываются в отдельный файл с расширением txt.

Запись в файл несет роль хранения данных программы, так как для этой цели не используем какую-либо базу данных, для удобства миграции проекта с данными на микрокомпьютер Raspberry.

В дальнейшем развитии программы предполагается использовать NoSQL базу данных Mongodb. Выбор данной базы данных обусловлен широкой поддержкой на языке программирования Python и возможности интеграции с веб-сервисами посредством программной платформы node.js с применением архитектуры взаимодействия REST (Representational state transfer).

Для визуализации данных в программе написан класс Paintgraf с использованием библиотеки Matplotlib. С помощью данного класса мы можем визуализировать плотности полученных данных на гистограммах, это может быть полезным для определения некой зависимости генерируемых данных и дает возможность понять подходящие ли генерируются параметры под нашу задачу.

На рисунке ниже изображена гистограмма значений параметра скорости:

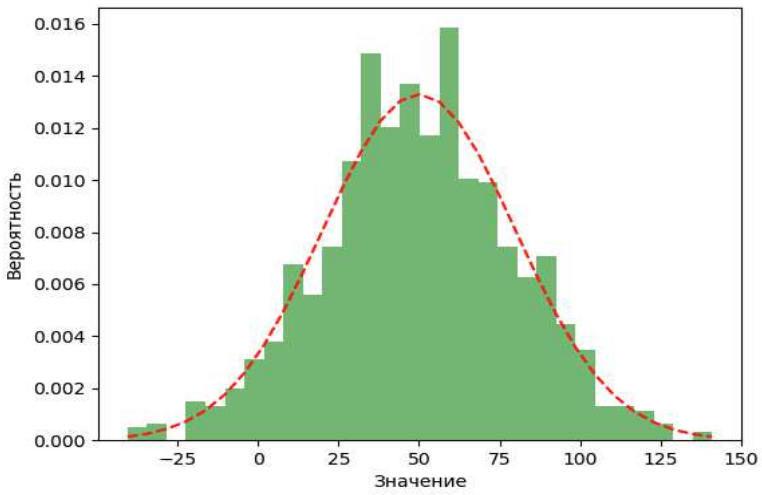

Рисунок 11 – Гистограмма значений параметра скорость

Для создания графика инициализируем экземпляр класса Paintgraf:

graf = grafs.Paintgraf()

```
graf.show_hist(speed, self.mu_speed, self.sigma_speed)
```

3.2.2 Обработка данных

Для обработки сгенерированных данных написан класс программы datamining.py

```
class Classifield:
    gen_data = []
    def __init__(self, data):
        self._data = data
        self._prepare_data()
    def _prepare_data(self):
        get_generate_data = open('files/gen_data.txt')
        l_gen_data = []
        for line in get_generate_data:
            l_gen_data.append(int(line))
        l_key = []
        l_val = []
        val_for_train = []
        val_for_train_learn = []
        cont_ok = 0
        cont_non = 0
        for i in self._data:
            l_key.append(int(i))
            l_val.append(self._data[i])
        n = 0
        for val in l_gen_data:
            r = functions.nearest(l_key, val)
            d_effect = self._data[str(r)]
            if d_effect > 88:
                cont_ok = cont_ok + 1
                val_for_train_learn.append(1)
            else:
```

```
            cont_non = cont_non + 1
            val_for_train_learn.append(0)
        val_for_train.append(val)
        n = n + 1
    get_generate_data.close()
```

В данном классе мы перебираем все полученные в классе генератора значения и сравниваем с значениями таблицы эффективности от производителя.

Сравнение происходит способом поиска ближайшего значения, то есть в цикле для каждого значения результата работы формулы подсчета выделения мощности для движения электромотоцикла происходит поиск ближайшего значения из таблицы производителя. При получении ближайшего значения мы можем взять процент КПД двигателя.

Далее производим классификацию жестким сравнением. Точкой сравнения взято значение равное 88% эффективности работы нашего электродвигателя. Данное значение относится к входным параметрам программы и взято как тестовое, так же оно зависит от требующегося уровня классификации эффективности.

При генерации параметров в количестве 50000 штук, получаем результат, продемонстрированный на диаграмме ниже:

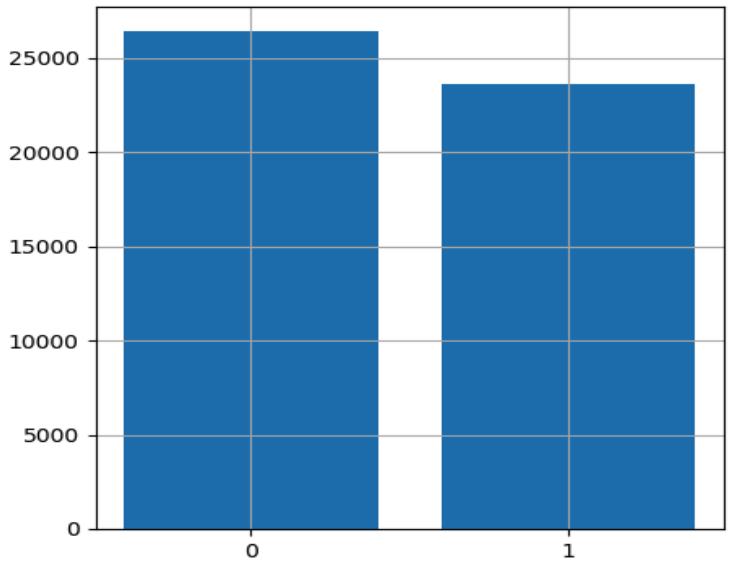

Рисунок 12 – Графический вывод результата классификации сравнением

Для создания диаграммы инициализируем экземпляр класса Paintgraf. По диаграмме видно, что больший процент набора параметров оказался неэффективным для нашей задачи. При повторении генерации в 50000 параметров при средней скорости в 50км/ч, в среднее значение уклона в 4 градуса, преимущественно по асфальту, при среднем значении ветра в 4 м/с получаем следующие результаты:

Таблица 2 – Результат классификации параметров для города

№ п/п	0 - неэффективно	1 - эффективно
1	26520	23479
2	26431	23568
3	26276	23723
4	26542	23457
5	26404	23595
6	26466	23533
7	26516	23483
8	26615	23384
9	26576	23423
10	26307	23692

Значения приведены в числовом виде для повышения точности наблюдения. Наблюдается закономерность, что распределение значения энтропии при данных исходных параметрах в большой степени равно большему проценту случаев неэффективного движения электромотоцикла.

Модель заданных диапазонов для генерации параметров симулировала езду по городу.

Проведем еще один эксперимент симулирую поездку с средней скоростью в 35 км/ч по холмистой местности с грунтовым покрытием и малой вероятностью порывов ветра.

Таблица 3 – Результат классификации параметров для холмистой местности

№ п/п	0 - неэффективно	1 - эффективно
1	17877	32122
2	17832	32167
3	17687	32312
4	17817	32182
5	17786	32213
6	17900	32099
7	17760	32239
8	18001	31998
9	17901	32098
10	17878	32121

В данном эксперименте наблюдается обратная зависимость – значения энтропии параметров распределяются таким образом, что больший % эффективного движения электромотоцикла при данных входящих параметров.

Исходя из данных двух экспериментов можно сделать вывод, что электромотоцикл с нашими исходными характеристиками больше подходит для поездок по холмистой местности с небольшой скоростью.

При проведении большего числа экспериментов с различными средними входными параметрами можно определить зависимости по другим моделям движения электромотоцикла.

3.2.3 Классификатор случайный лес

В Главе 2 данной работы были рассмотрены методы классификации данных с применением машинного обучения. Более подходящий метод к нашей работе — это метод «Случайный лес».

Для работы с методом случайный лес написан класс Randforest с применением библиотеки sklearn.

```
class randforest:
    def __init__(self, data_x, data_y):
        self.data_x = data_x
        self.data_y = data_y
        self._random_forest()
    def _random_forest(self):
        get_generate_angle = open('files/gen_data_angle.txt')
        get_generate_fric = open('files/gen_data_fric.txt')
        get_generate_veter = open('files/gen_data_veter.txt')
        x_train = []
        x = 0
        with open('files/gen_data_speed.txt') as f:
            for speed in f:
                if x <= len(self.data_x)-1:
                    angle = float(get_generate_angle.readline())
                    fric = float(get_generate_fric.readline())
```

```
                veter = float(get_generate_veter.readline())
                x_train.append([float(speed), angle, fric, veter])
            x = x + 1
        # массив вида [1, 1, 0, 0]
        y_train = self.data_y
        clf_rf = RandomForestClassifier()
        X_train, X_test, Y_train, Y_test = train_test_split(x_train, y_train,
test_size=0.2)
            clf_rf.fit(X_test, Y_test)
            clf_rf.predict(X_train)
```

В данном классе мы берем все сохраненные сгенерированные параметры, в данном решении по 50000 значений каждого параметра и применяем их к функции RandomForestClassifier() библиотеки sklearn. Данная функция содержит в себе исполнение метода «Случайный лес».

Для обучающей выборки поделим объясняющие параметры пополам для тестовых данных и проверяющих, для этого в sklearn есть функция train_test_split(). Обучение происходит с помощью функции fit() и само предсказание по входящим параметрам инициализируется с помощью функции predict().

После обучения и запуска на проверочных данных получим результат работы класса в виде массива значений такого вида:

[1 0 0 1 1 0 … 1 0 0 0 1]

То есть, все параметры предсказаны для возможного класса эффективности.

Для более подробной аналитики результата работы метода воспользуемся функцией библиотеки sklearn для оценки вероятности присваивания класса - predict_proba(x).

Результат работы данной функции является массив следующего вида:

[0.4 0.6]

[0.6 0.4]

[0.4 0.6]

[0.5 0.5]

[0.3 0.7]

[0. 1.]

[0.7 0.3]

[0.2 0.8]

[0.6 0.4]

[0.4 0.6]

[0.3 0.7]

Можно проследить с какой вероятностью метод отнес выбранный набор входящих данных к классам эффективно и неэффективно. Подсчитав количество результатов с вероятность более 80% точности получили 8-15% таких решений от обшей выборки. Чтобы повысить качество работы метода были внесены дополнительные параметры к экземпляру класса RandomForestClassifier:

Criterion - критерий для выбора решающих правил при построении узлов деревье, выбран вариант «entropy». Max_depth - максимальная глубина дерева. Выбран вариант None, то есть дерево строится либо пока каждый лист не содержит объектов из разных классов, либо пока лист содержит не более min_samples_split (количество признаков для деления) объектов выборки.

Min_samples_split - минимальное количество объектов выборки, необходимое для разделения узла дерева и образования решающего правила, выбран вариант «2».

N_estimators - количество решающих деревьев в композиции, выбран вариант «10».

Графическое представление результата работы метода:

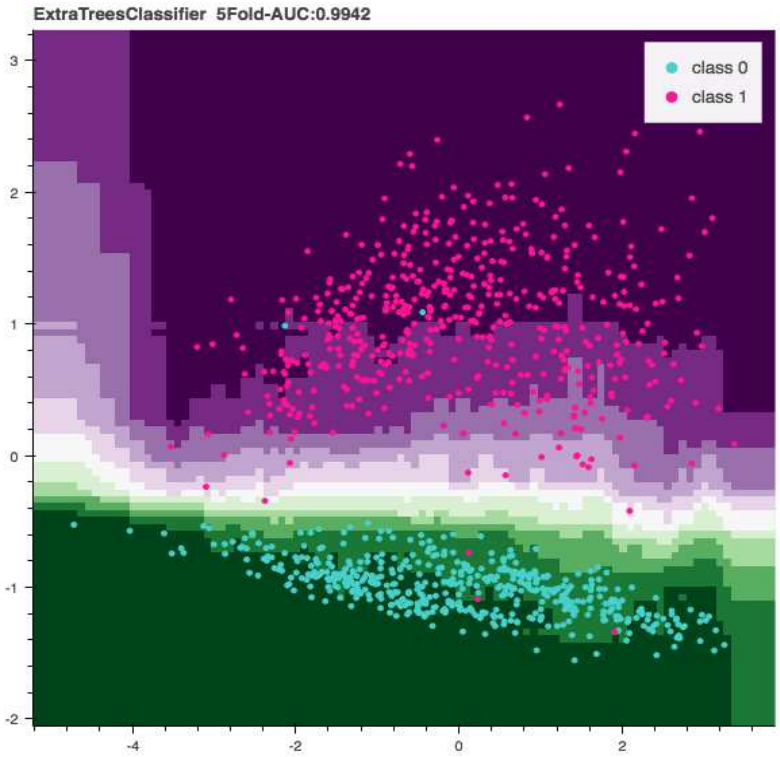

Рисунок 13 - Градиент результата классификации

Данные параметры улучшили качество работы метода до 35-45% более высокой вероятности (более 80% правильности классифицирования).

Чтобы можно было наблюдать наиболее важный для классификации в данном ансамбле параметр используем функцию feature_importances_

Результат ее работы представим в виде графика:

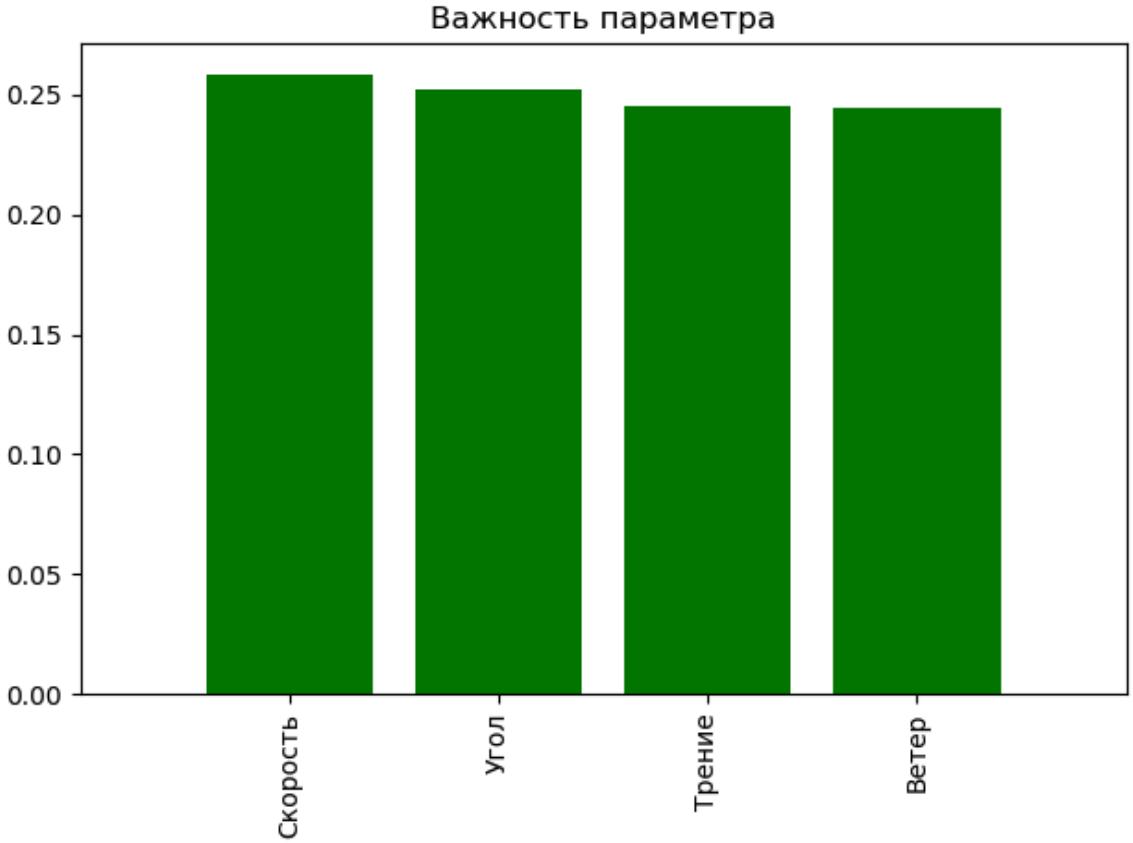

Рисунок 14 – Важность параметров метода Случайный лес

В числовом виде это выглядит так: [0.25815397 0.2520947 0.24524019 0.24451114].

По данному результату работы функции можно сделать вывод, что значение скорости влияет больше на результат классификации по сравнению с другими объясняющими параметрами.

Чтобы сохранить результат обучения метода воспользуемся функцией библиотеки sklearn joblib.dump(), файл сохраняется в формате pkl.

Для загрузки сохраненного результата в проект воспользуемся функцией библиотеки sklearn joblib.load().

3.2.4 Получение данных с датчиков

Для получения входных данных датчиков написан набор функций в файле Gpioinput с применением библиотеки RPi.GPIO

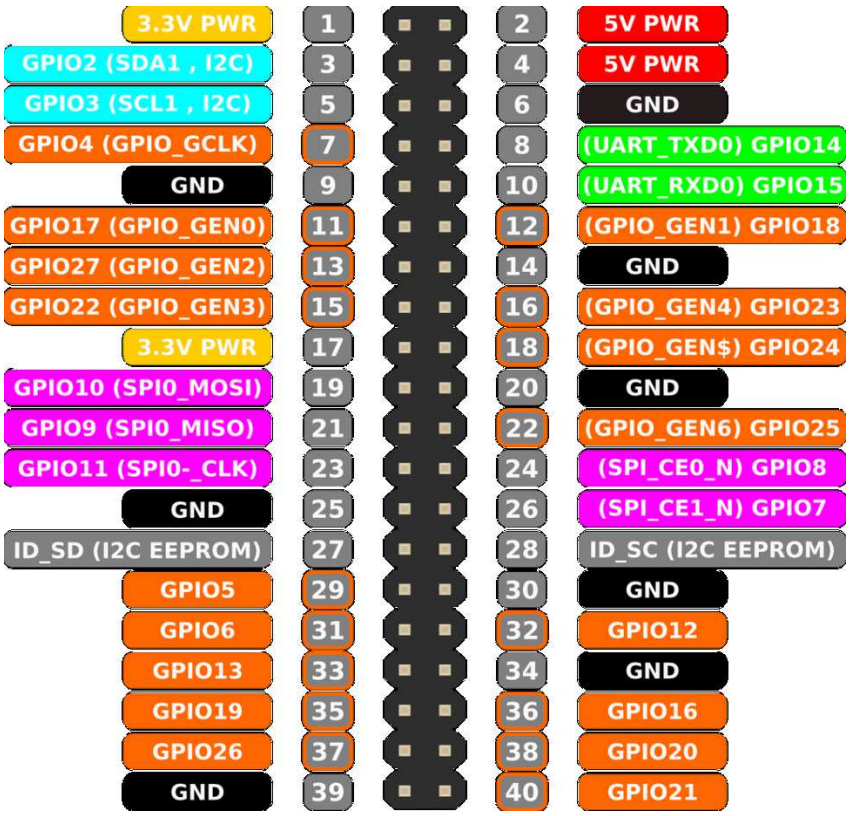

Рисунок 15 – Распиновка Raspberry PI 3

Для получения входных данных с датчиков написан функционал с использованием библиотеки RPi.GPIO, данная библиотека написана специально для Raspberry PI и работает с ее пинами.

Для того, чтобы дать понять какой пин слушать в данное время требуется установить для данного пина флаг IN: GPIO.setup(номер пина, GPIO.IN).

Для работы программы требуется получать данные с датчика скорости – датчик Холла, датчик гироскопа, датчик ветра, коэффициент трения дороги задается внутри программы с амплитудой погрешности.

Также все входящие данные переводятся в систему исчисления, используемую в формуле расчета требуемой мощности для движения электромотоцикла.

Например, используя датчик Холла мы получаем угловую скорость вращения колеса, для нахождения скорости требуется умножить полученное значение датчика на число П и на радиус колеса электромотоцикла.

Тестирование

Написанная программа может быть разделена на две части – первая для моделирования получения эффективности из сгенерированных значений и обучения метода машинного обучения классификации. Вторая – для получения входных данных с датчиков. Основываясь на сохраненном опыте, полученном в первой части модуля, позволяет проводить аналитический вывод о текущей эффективности при движении электромотоцикла.

Для тестирования программы были выбраны параметры входных данных для генерации 50000 значений подходящие под условия города – это средняя скорость 50 км/ч, средний угол наклона дороги 4%, коэффициент трения асфальтовой дороги 0,007, среднее значение скорости ветра 4 м/с.

И параметры для холмистой местности – средняя скорость 35 км/ч, средний угол наклона дороги 8%, коэффициент трения грунтовой дороги 0,025, среднее значение скорости ветра 2 м/с.

Провели по 10 экспериментов генерации данных, была выявлена зависимость в результатах.

Результаты первоначальной классификации (эффективно и неэффективно) приведены в таблицах 2 и 3.

Из таблиц можно сделать вывод, что при классификации больше вероятностей, что эффективен электромотоцикл будет при езде по холмистой местности чем по городской среде.

Для обучения метода машинного обучения «Случайный лес» отсеяно 20% от общего количества сгенерированных параметров – это 10000 параметров. Оставшиеся 40000 параметров были задействованы для классификации данным методом. По результатам прогона получили 17011 значений с наиболее высокой энтропией. Результат обучения сохранен в файл с расширением pkl.

Также можно проследить на графике, какой из параметров больше участвовал в классификации метода, для этого нужно обратиться к Рисунку 14.

Вторая часть модуля, которая состоит из функций принятия входных данных с датчиков и классификации их, основываясь на обученной части выборки из pkl файла, установлена на микрокомпьютер Raspberry PI. При получении параметров строим по ним автообновляемую гистограмму и классифицируем с коэффициентом энтропии.

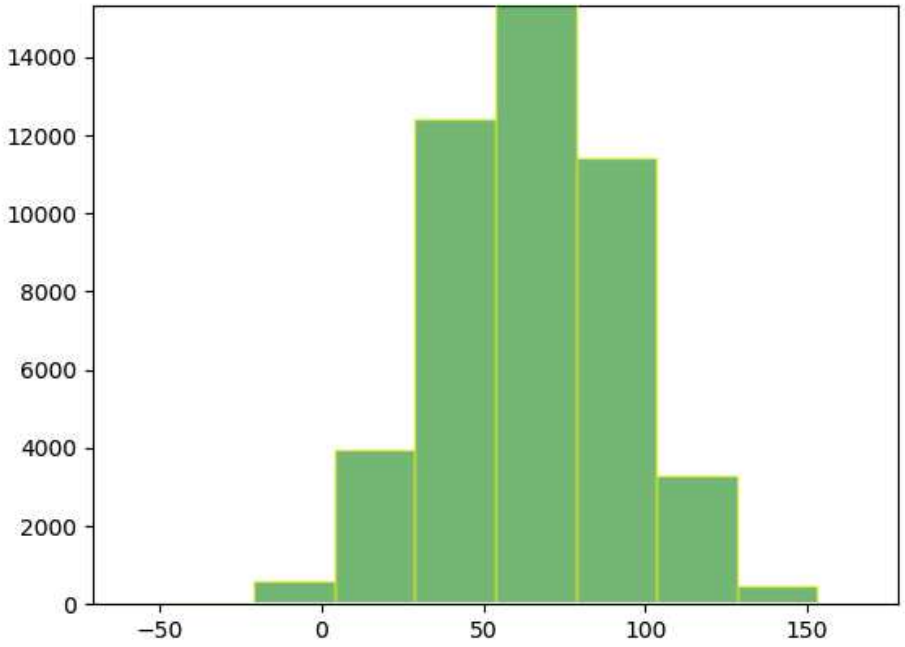

Рисунок 16 - Гистограмма входящего параметра

Тестирование проведено, модули программы работают успешно.

Выводы по главе 3

Разработана программа для информационно-аналитического модуля электромотоцикла. Проведено её успешное тестирование. На основании полученных результатов выполнения работы данного модуля можно сделать аналитические выводы, которые служат основанием для изменения настроек электромотоцикла для более эффективного расхода запаса батареи при прохождении пути.

Дальнейшее использование полученных результатов работы данного модуля представляет из себя несколько вариантов:

1) Настройка контроллера батареи.

2) Корректировка выбранного маршрута.

3) Аналитика для водителя электромотоцикла.

Настройка контроллера батареи позволяет заранее, перед движением, задать такие параметры как:

а) Рекуперативное торможение (Да/Нет).

б) Номинальное напряжение, В.

в) Батарейный ток, А.

г) Пиковый, фазный ток, А.

д) Максимальная частота вращения, об/мин.

е) Ускорение, %.

При корректировки выбранного пути можно подобрать такой путь, который бы показывал при тестировании сгенерированных параметров, основывавшихся на выбранном пути, наилучший результат. Например, если требуется запустить электромотоцикл как пассажирский транспорт пассажира с института на улице Свободный проспект в институт на улице Академика Киренского, то можно сгенерировать параметры с средним показателем данного маршрута и по полученным результатам прохождения метода классификации сделать выводы для дальнейшего решения по настройке электромотоцикла.

Аналитика для водителя представляет из себя гистограмму на дисплее:

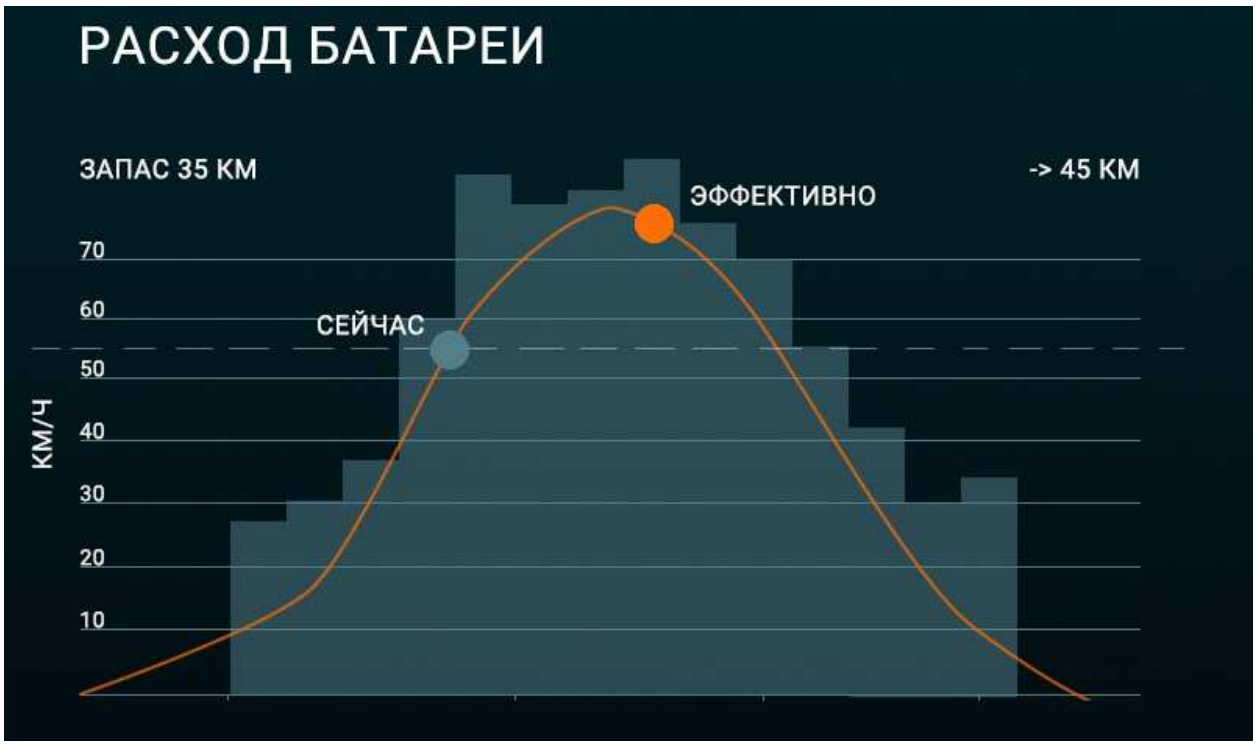

Рисунок 17 – Концепт аналитического спидометра

Данный вид альтернативы спидометра позволит водителю понимать ситуацию эффективности расхода батареи. Данный вид графика позволит сравнить текущее положение одного из входящих параметров и по отношению к другим и основываясь на обучении машинного обучения метода «Случайный лес» - данная система выведет на экран более эффективную комбинацию параметров для увеличения запаса хода электромотоцикла.

Так, основываясь на полученные аналитические данные, мы не можем напрямую влиять на ход электромотоцикла, так как это может привести у аварийной ситуации, то предложен вариант разработки и установки на электромотоцикл механизма - вариатора для автоматической регулировки момента электродвигателя и его оборотов.

ЗАКЛЮЧЕНИЕ

Проанализирована актуальность рассматриваемой темы в данной работе, изучены другие статьи и работы по темам оптимизации электропотребления. Объектом исследования выбран электромотоцикл, построенный на базе СФУ, как объект, для которого требуется разработать информационно-аналитический модуль для выполнения задачи оптимизации потребления энергии батареи, что позволит проезжать на большие расстояния при исходных характеристиках двигателя, конструкции и дизайна.

Для решения данной задачи рассмотрены методы классификации, выбран наиболее подходящий – метод «Случайный лес». Выбран язык программирования Python и микрокомпьютер Raspberry PI как вычислительный блок.

Разработана программа для модуля на выбранном языке программирования, которая содержит в себе часть для моделирования с помощью генерации подобранных параметров и часть обработкой входящих данных с датчиков, установленных на электромотоцикле. Успешно протестирована работа программы, выявлены закономерности и качество полученных результатов, предложены варианты дальнейшего использования и развития разработанного модуля.

СПИСОК СОКРАЩЕНИЙ

ИА – информационно-аналитический

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Электронный научный журнал [Электронный ресурс] : методы улучшения характеристик батареи. – Режим доступа: https://futurism.com
2. Крицкий С.Д., Купреев С.А., Тарасов А.В. Проект электромотоцикла категории L3 – публикация в сборнике международной научно-практической конференции в г. Сочи, 2016г.
3. Крицкий С.Д. Обработка данных в задачах оптимизации расхода электроэнергии – статья в сборнике конференции Робототехника и искусственный интеллект, г. Железногорск, 2017г.
4. Представитель производителя электродвигателей в России [Электронный ресурс] : таблица эффективности электродвигателя от производителя. – https://goldenmotor.ru/bldc-motors/hpm5000b-fan-cooling/
5. Караичев К. Н., Зырин В. О. Повышение эффективности работы электроцентробежных насосов на месторождениях с трудноизвлекаемыми запасами нефти. 2016.
6. Мин Т. А., Суздорф В. И. Энергоэффективное управление двигателем последовательного возбуждения. 2017.
7. Борисевич А. В. Практический алгоритм численного поиска для минимизации мощности потерь асинхронного электродвигателя. 2014.
8. Золотых С.Ф., Рожков С.В., Лобанова С.В. Анализ методов повышения энергоэффективности электродвигателей в машиностроении. 2013.
9. Милых В.И., Ткаченко С.В. Анализ влияния параметров линейного импульсного электродвигателя на эффективность его работы.
10. Лысова О.А., Портнягин А.Л., Кочнев П.В. Разработка и исследование энегосберегающей системы управления регулируемыми электродвигателями насосов кнс.
11. Калмыков А.Н, Галушин С.Я., Дмитриев Б.Ф. Система управления электродвигателем гребной электрической установки.

12. Савоськин А. Н., Телегин М. В., Плис В. И., Жирков А. И. Предложения по улучшению системы автоматического управления током тяговых электродвигателей электровоза для обеспечения его работоспособности.
13. Овсянников Е.М., Нгуен К.Т. Система прямого управления моментом и потокосцеплением ротора асинхронного электродвигателя.
14. Таранов Д.М., Каун О.Ю., Гетманенко В.М., Лебедев К.Н., Литвинов В.Н., Старовойт М.Ю. Имитационная модель системы «преобразователь частоты - асинхронный электродвигатель» насосов системы водоснабжения с адаптивным алгоритмом управления.
15. Яночкина О. О. Об исследовании динамики систем управления электродвигателем.
16. Москера М. М. Моделирование систем автоматического управления электродвигателя робота-конвертоплана.
17. Максаков С.А. Проектирование системы автоматического управления электродвигателем постоянного тока.
18. Минакова Т.Е. Параметры режимов асинхронных электродвигателей в системе умного управления.
19. Кривилев А.В. Разработка и реализация на плис энергоэффективных способов импульсного управления системами "усилитель мощности - электродвигатель" на основе методов автоматизированного проектирования.
20. Захаров В.С. Цифровые системы управления асинхронным электродвигателем.
21. База статей [Электронный ресурс] : методы классификации данных. – Режим доступа: http://machinelearning.ru
22. Информационные технологии [Электронный ресурс] : обзор языка программирования Python для математических операции. – Режим доступа: https://habr.com/post/312268/

23. Официальный сайт документация дистрибутива Anaconda [Электронный ресурс] : информация и документация о дистрибутиве. – Режим доступа: https://anaconda.org

Федеральное государственное автономное образовательное
учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
Кафедра систем искусственного интеллекта

УТВЕРЖДАЮ
Заведующий кафедрой
_____ Г. М. Цибульский
подпись
«__» _____ 2018 г.

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Разработка информационно-аналитической модели для блока управления электромотоцикла.

09.04.02 Информационные системы и технологии

09.04.02.01 Информационно-управляющие системы

Научный руководитель	_____ подпись, дата	профессор, д-р. ф.-м. н.	Б. С. Добронец
Выпускник	_____ подпись, дата		С. Д. Крицкий
Рецензент	_____ подпись, дата	профессор, д-р. ф.-м. н.	В. Д. Кошур
Нормоконтролер	_____ подпись, дата	профессор, д-р. ф.-м. н.	Б. С. Добронец

Красноярск 2018