

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»

Высшая школа электроники и компьютерных наук
Кафедра «Электронные вычислительные машины»

ДОПУСТИТЬ К ЗАЩИТЕ
Заведующий кафедрой ЭВМ
_____ Д.В. Топольский
« ___ » _____ 2024 г.

Использование возможностей нейросетей для оптимизации параметров
вентильного двигателя постоянного тока

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ
ЮУРГУ-090301.2024.405 ПЗ ВКР

Руководитель работы,
д.т.н., профессор, профессор каф.
ЭВМ
_____ С.А. Ганджа
« ___ » _____ 2024 г.

Автор работы,
студент группы КЭ-405
_____ В. Е. Романов
« ___ » _____ 2024 г.

Нормоконтролёр,
ст. преп. каф. ЭВМ
_____ С.В. Сяськов
« ___ » _____ 2024 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»
Высшая школа электроники и компьютерных наук
Кафедра «Электронные вычислительные машины»

УТВЕРЖДАЮ
Заведующий кафедрой ЭВМ
_____ Д.В. Топольский
«__» _____ 2024 г.

ЗАДАНИЕ
на выпускную квалификационную работу бакалавра
студенту группы КЭ-405
Романову Владиславу Евгеньевичу,
обучающемуся по направлению
09.03.01 «Информатика и вычислительная техника»

1. Тема работы: «Использование возможностей нейросетей для оптимизации параметров вентильного двигателя постоянного тока» утверждена приказом по университету от 22 апреля 2024 г. №764-13/12

2. Срок сдачи студентом законченной работы: 01 июня 2024 г.

3. Исходные данные к работе

Проектируемая система должна соответствовать следующим функциональным требованиям:

- система должна считывать данные пользователя из файла формата «.csv»;
- система должна предоставлять функционал для предварительной обработки данных, включая очистку, преобразование и масштабирование;
- система должна обучаться на полученных данных;
- система должна в результате показать графики предсказанными значениями;
- система должна предоставить функционал для ввода мощности двигателя и вывести предсказанные коэффициенты для параметров двигателя.

Проектируемая система должна удовлетворять следующим нефункциональным требованиям.

- система должна обеспечивать высокую скорость обучения и предсказания модели, чтобы пользователи могли быстро получать результаты;
- система должна быть реализована на языке Python.

4. Перечень подлежащих разработке вопросов

1. Провести анализ предметной области и обзор литературы.
2. Определение требований к разрабатываемой системе.
3. Реализовать и обучить модели на основе нейронных сетей.
4. Проанализировать полученные данные, сравнить модели и использовать полученные данные обобщенных переменных для инженерной методики проектирования ВЭКВ.

5. Дата выдачи задания: 1 декабря 2023 г.

Руководитель работы _____ / С.А.. Ганджа/

Студент _____ / В.Е. Романов/

КАЛЕНДАРНЫЙ ПЛАН

Этап	Срок сдачи	Подпись руководителя
Введение и обзор литературы	01.03.2023	
Анализ требований к проектируемой системе	01.04.2023	
Разработка моделей нейронной сети	01.05.2023	
Обучение и сравнение результатов обучения моделей нейронной сети	15.05.2023	
Компоновка текста работы и сдача на нормоконтроль	24.05.2023	
Подготовка презентации и доклада	30.05.2023	

Руководитель работы _____ / *С.А.Ганджа* /

Студент _____ / *В.Е.Романов* /

АННОТАЦИЯ

В.Е. Романов. Использование возможностей нейросетей для оптимизации параметров вентильного двигателя постоянного тока. – Челябинск: ФГАОУ ВО «ЮУрГУ (НИУ)», ВШ ЭКН; 2024, 70 с., 26 ил., библиогр. список – 24 наим.

В рамках выпускной квалификационной работы производится проектирование моделей нейронных сетей для поиска оптимальных обобщенных переменных. Процесс выполнения выпускной квалификационной работы включает в себя решение следующих вопросов: анализ предметной области и обзор литературы, определение требований к разрабатываемой системе, реализация и обучение моделей нейронных сетей, анализ обучения нейронных сетей, использование полученных данных обобщенных переменных для инженерной методики проектирования вентильного электродвигателя комбинированного возбуждения.

В работе рассмотрены преимущества вентильного электродвигателя комбинированного возбуждения, продемонстрирован метод поиска оптимальных гиперпараметров нейронной сети и методика расчета внутренней геометрии вентильного двигателя с помощью обобщенных переменных.

Основная цель разрабатываемой нейронной сети – упростить и ускорить проектирование геометрии вентильного двигателя.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	7
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	11
1.1. Описание предметной области	11
1.2. Конструкция и принцип действия ВЭКВ.....	11
1.3. Разработка математической модели ВЭКВ на основе схем замещения.	12
1.4. Обзор научной литературы.....	15
1.5. Сравнительный анализ аналогов	19
2. АНАЛИЗ ТРЕБОВАНИЙ К ПРОЕКТИРУЕМОЙ СИСТЕМЕ	20
2.1. Функциональные и нефункциональные требования.....	20
2.2. Диаграмма вариантов использования.....	20
3. РЕАЛИЗАЦИЯ	25
3.1. Программные средства реализации	25
3.2. Подготовка данных.....	26
3.3. Реализация модели многослойного перцептрона.....	33
3.4. Реализация модели рекуррентной нейронной сети.....	39
3.5. Реализация функций предсказания и построения графиков.....	43
4. ЭКСПЕРИМЕНТАЛЬНАЯ ЧАСТЬ	46
4.1. Сравнение предсказанных значений на графиках	46
4.2. Тестирование функции предсказания	49
4.3. Инженерная методика проектирования ВЭКД на основе коэффициентов	50
4.4. Расчет магнитной цепи	56
4.5. Расчет обмотки возбуждения	60
4.6. Расчет массы активных материалов.....	62
4.7. Расчет потерь в зубцах	63
ЗАКЛЮЧЕНИЕ	67
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	68

ВВЕДЕНИЕ

Вентильный двигатель постоянного тока [14] (ВДПТ) - это электрический двигатель, который использует постоянный ток для создания вращающегося движения. Он состоит из статора и ротора, которые взаимодействуют для создания вращения.

Работа ВДПТ основана на принципе электромагнитной индукции. Когда постоянный ток подается на статор, создается магнитное поле. Ротор, который содержит постоянные магниты или электромагниты, взаимодействует с этим магнитным полем и начинает вращаться. Пример работы показан на рисунке 1.

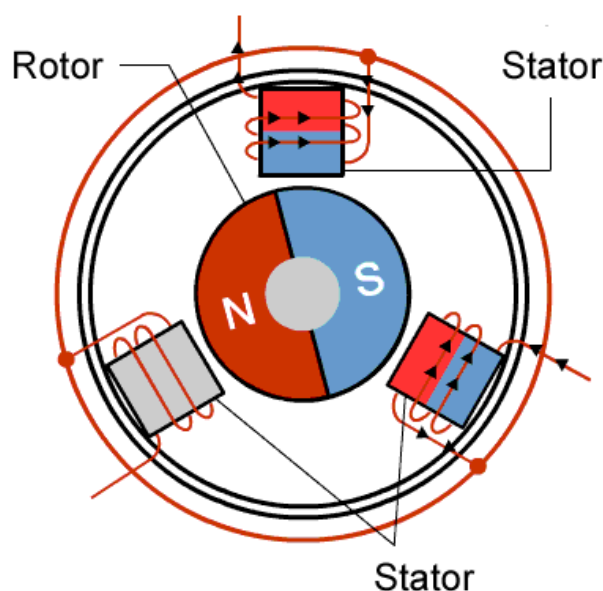


Рисунок 1 – пример работы ВДПТ

Вентильные двигатели постоянного тока обладают высокой эффективностью, высоким крутящим моментом и широким диапазоном скоростей вращения. Они широко используются в различных промышленных и бытовых приложениях, таких как электроника, автомобильная промышленность и приводы для механизмов.

Актуальность

Вентильные двигатели постоянного тока находят широкое применение в различных областях, где требуется высокая точность позиционирования и

контроля скорости вращения. Некоторые из основных областей применения ВДПТ включают:

1. Промышленная автоматизация: ВДПТ используются в промышленных роботах, автоматических линиях сборки, оборудовании для обработки материалов и других системах автоматизации для точного управления движением.

2. Медицинское оборудование: ВДПТ применяются в медицинском оборудовании, таком как медицинские сканеры, дентальные установки и оборудование для хирургических операций, где требуется высокая точность и надежность.

3. Оптические системы: ВДПТ используются в оптических системах, таких как телескопы, микроскопы и лазерные системы, для точного управления положением оптических элементов.

4. Авиационная и космическая промышленность: ВДПТ применяются в системах управления положением и стабилизации на борту самолетов, космических аппаратах и спутниках.

Использование нейронных сетей при разработке ВДПТ позволяет создавать более интеллектуальные и адаптивные системы управления и мониторинга, что в свою очередь способствует повышению их производительности, надежности и эффективности.

Постановка задачи

При проектировании вентильных электродвигателей следует учитывать большое количество исходных данных. Помимо данных технического задания, где указана номинальная мощность, номинальное напряжение, номинальная частота вращения при расчете необходимо учитывать разнообразные характеристики электротехнических материалов (кривые намагничивания стали, электропроводность обмоточной меди в зависимости от температуры, различные варианты конструктивных исполнений, влияние на параметры и характеристики электронной системы управления). Набор этих исходных данных представляет собой очень большой массив. Выбрать из этого массива оптимальное сочетание параметров представляет собой сложную задачу.

Нейронные сети, как раз достаточно хорошо справляются с задачей по обработке больших массивов исходных данных, выбирая при этом их наилучшее с точки зрения критериев качества сочетание.

В ЮУрГУ в течении ряда лет ведутся работы по оптимальному проектированию вентильных двигателей комбинированного возбуждения с применением, так называемых обобщенных переменных. За это время накопилась статистика, позволяющая реализовать проекты для конкретных технических заданий. Но количество электротехнических материалов постоянно растет, изменяются режимы системы управления. В этой ситуации можно воспользоваться возможностями нейронных сетей по обработке и оптимальному выбору данных для проектирования электродвигателей этого класса.

Таким образом целью ВКР является использование возможностей нейросетей для оптимального проектирования вентильных двигателей. В качестве конструктивной модификации выбран вентильный электродвигатель комбинированного возбуждения. План по расчету внутренней геометрии вентильного двигателя показан на рисунке 2.

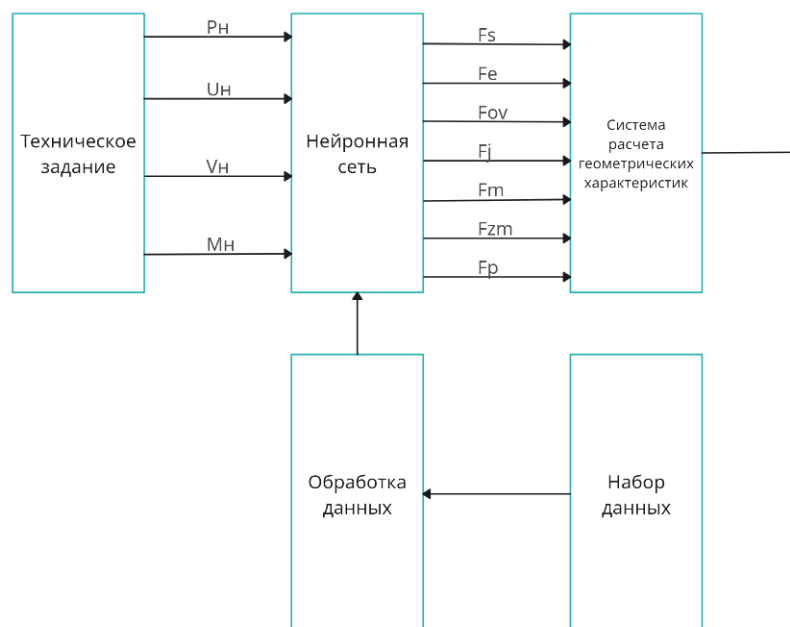


Рисунок 2 – План по расчету геометрии

Для выполнения этой цели необходимо последовательно решить следующие задачи:

- 1) выполнить анализ предметной области, выполнить обзор научной литературы;
- 2) выполнить анализ требований к проектируемой системе;
- 3) разработать модели на основе нейросетей;
- 4) использовать результаты нейросети по определению оптимальных значений обобщенных переменных для инженерной методики проектирования ВЭКВ.

Структура и содержание работы

Работа состоит из введения, четырех глав, заключения и списка литературы. Объем работы составляет 70 страниц, объем списка литературы – 24 источника.

В первой главе был выполнен анализ предметной области и обзор научной литературы.

Во второй главе был выполнен анализ требований к проектированию системы.

В третьей главе были разработаны различные модели нейронной сети.

В четвертой главе были проанализированы полученные результаты обучения и проведено сравнение различных моделей нейронных сетей. Были использованы полученные обобщенные переменные для инженерной методики проектирования ВЭКВ.

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. Описание предметной области

В данной выпускной квалификационной работе предлагается использование вентильного электродвигателя комбинированного возбуждения (ВЭКВ).

Продемонстрируем преимущества этого двигателя по сравнению, например, с двигателем, который использует только постоянные магниты для возбуждения.

Частоту вращения вентильного двигателя n можно определить по следующей формуле:

$$n = \frac{U - I_a \cdot R_a}{C_E \cdot \Phi_\delta}, \quad (1.1)$$

где U – напряжение питания; I_a – ток якоря; R_a – активное сопротивление цепи якоря; C_E – машинная постоянная; Φ_δ – рабочий магнитный поток.

Вентильная машина с постоянными магнитами не может изменять свой рабочий магнитный поток Φ_δ . Изменение результирующего магнитного потока возможно только за счет смещения коммутации с нейтрального положения, что приводит к потерям мощности. Вентильная машина с комбинированным возбуждением позволяет регулировать частоту вращения как через цепь якоря, так и через цепь возбуждения, что обеспечивает широкий диапазон регулирования. Из теории известно, что регулировка по цепи возбуждения является энергетически очень выгодной, т. к. через неё проходит всего 1.0-5.0 % потребляемой мощности двигателя.

Комбинация возбуждения от обмотки и магнита дает следующие преимущества. При электромагнитном возбуждении рабочий магнитный поток Φ_δ зависит:

- 1) от площади поперечного сечения магнитопровода;
- 2) от площади поперечного сечения обмотки возбуждения.

Это означает, что рабочий магнитный поток зависит от линейных размеров a в 4-той степени.

$$\Phi_\delta \equiv a^4 \quad . \quad (1.2)$$

При возбуждении от постоянных магнитов магнитный поток Φ_δ зависит от объема, то есть от линейных размеров a в третьей степени. Таким образом, если заменить часть магнитного потока постоянными магнитами, можно значительно уменьшить габариты. Использование обмотки возбуждения позволяет регулировать магнитный поток и упрощает электронную аппаратуру, поскольку суммарный поток Φ_Σ формируется, как результат сложения потока обмотки возбуждения $\Phi_{ОВ}$ и потока постоянных магнитов $\Phi_{ПМ}$.

$$\Phi_\Sigma = \Phi_{ПМ} + \Phi_{ОВ}. \quad (1.3)$$

Исходя из этого, применение машин с комбинированным возбуждением позволяет уменьшить габариты за счет мощных высококоэрцитивных магнитов и обеспечить широкий диапазон регулирования благодаря использованию обмотки возбуждения

В ВЭКВ регулирование крутящего момента и частоты вращения осуществляется как путем изменения величины питающего напряжения, так и путем регулирования магнитного потока, создаваемого обмоткой возбуждения. Этот принцип позволяет достичь широкого диапазона регулирования скорости и момента при относительно малой массе и габаритах. Применение высококоэрцитивных постоянных магнитов способствует снижению массы и объема двигателя.

1.2. Конструкция и принцип действия ВЭКВ

Статор ВЭКВ состоит из шихтованных пакетов 1 и 2, содержащих пазы для якорной обмотки, которые размещены отдельно в осевом направлении и установлены на массивные части магнитопровода 4 и 5. Магнитопровод может быть двухчастным, при этом части соединяются со втулкой 6. Возможно конструктивное исполнение, когда одна часть магнитопровода интегрирована с втулкой. Между массивными частями магнитопровода расположена обмотка возбуждения 7, которая создает магнитный поток для всех полюсов благодаря своей намагничивающей силе. Ротор представляет собой магнитопровод в форме кольца и размещен снаружи статора. В кольце расположены постоянные

магниты 9 и магнитопроводящие вкладыши 10. На одной стороне магнитопровода постоянные магниты одной полярности чередуются с вкладышами, а на другой стороне - магниты противоположной полярности чередуются с вкладышами. Между этими кольцами имеется воздушный зазор [16]. На рисунке 3 изображен эскиз двигателя.

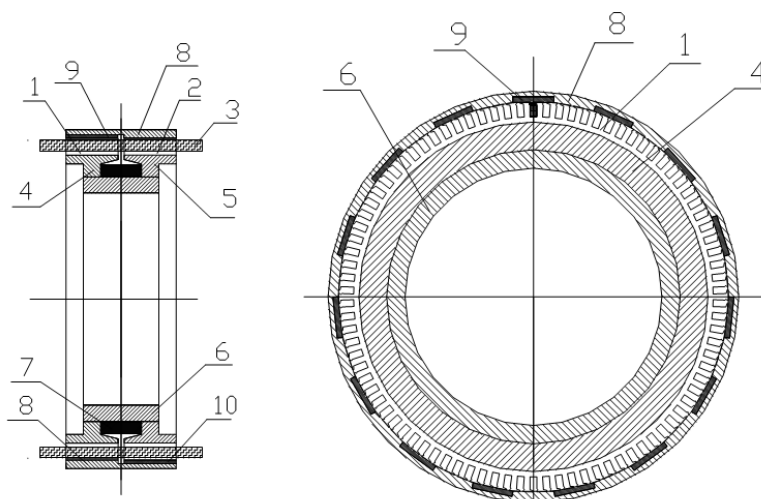


Рисунок. 3 Эскиз двигателя

Постоянные магниты в первом кольце сдвинуты относительно постоянных магнитов во втором кольце на одно полюсное деление вдоль оси вращения.

Принцип работы машины комбинированного возбуждения данной конструкции следующий. Магнитопровод содержит два источника магнитного поля: постоянные магниты и обмотку возбуждения. Оба источника создают свои магнитные потоки, которые замыкаются по разным маршрутам.

Магнитный поток от постоянных магнитов проходит следующим путем: через первый воздушный зазор, затем через зубцовую зону первого пакета статора, далее через спинку первого пакета статора, далее через массивный магнитопровод, затем через спинку второго пакета статора, далее через зубцовую зону второго пакета статора, далее через второй воздушный зазор, затем через магнитопроводящий вкладыш второго кольца ротора, далее через

спинку ротора и наконечник через магнитопроводящий вкладыш первого кольца ротора.

Магнитный поток от обмотки возбуждения проходит следующим путем: через первый воздушный зазор, затем через зубцовую зону первого пакета статора, далее через спинку первого пакета статора, затем через массивный магнитопровод, далее через спинку второго пакета статора, затем через зубцовую зону второго пакета статора, далее через второй воздушный зазор, и наконец через постоянный магнит с одной полярностью второго кольца ротора, далее через спинку ротора и постоянный магнит с противоположной полярностью второго кольца ротора.

Таким образом, магнитные потоки пересекаются только в спинке якоря и спинке индуктора. При этом магнитный поток от постоянных магнитов остается неизменным по величине и направлению, в то время как магнитный поток от обмотки возбуждения может изменять как величину, так и полярность. Это позволяет регулировать суммарный магнитный поток за счет обмотки возбуждения.

Оба магнитных потока при вращении ротора пересекают обмотку статора, индуцируя в ней ЭДС и токи. В результате взаимодействия тока якоря и результирующего магнитного потока от обоих источников возникает электромагнитный момент.



Рисунок 4 – Схема подключения обмоток

Схема подключения обмоток, показанная на рисунке 4, предполагает подключение обмотки возбуждения к мостовой схеме регулятора напряжения.

Данная конструкция позволяет с помощью регулирования потока от обмотки возбуждения обеспечить широкий диапазон регулирования скорости и момента ВЭКВ [16, 17].

1.3. Разработка математической модели ВЭКВ на основе схем замещения.

Геометрия магнитной цепи в данной конструкции отличается от традиционных вентильных машин. Эта специфика требует учета при разработке методики расчета.

Высококоэрцитивные магниты обладают высоким магнитным сопротивлением, сравнимым с магнитным сопротивлением воздуха. Из-за этого постоянные магниты не пропускают внешние магнитные поля через себя, когда создают собственный магнитный поток. Этот факт объясняет разделение магнитных потоков от постоянных магнитов и от обмотки возбуждения по различным маршрутам. Следовательно, магнитный поток от постоянного магнита индуцирует ЭДС только в той части проводника, которая находится под постоянным магнитом, в то время как в другой части, расположенной под вкладышем, ЭДС наводится от обмотки возбуждения. Суммарная ЭДС определяется как сумма ЭДС от постоянных магнитов и от обмотки возбуждения.

При расчете магнитной цепи необходимо учитывать эту особенность и проводить расчет двух магнитных цепей с соответствующими магнитными потоками.

В участках магнитной цепи, где геометрия соответствует стандартным машинам, расчет проводится с использованием известных традиционных методик [18-23]. Для участков магнитной цепи с отличающейся геометрией необходимо применять предложенную методику расчета.

Все уравнения расчета должны быть объединены в единую математическую модель. Эта модель будет использоваться в оптимизационных циклах для выбора оптимальной геометрии по заданным критериям. Поэтому математическая модель

должна быть достаточно простой и позволять многократные расчеты в процессе оптимизации.

Наиболее подходящим способом для создания такой математической модели может быть метод схем замещения [24]. Этот метод достаточно прост и позволяет включать математическую модель в многочисленные циклы перебора при оптимизации. Погрешность расчета с использованием этого метода определяется точностью задания большого числа коэффициентов. Качество математической модели можно улучшить, если значения этих коэффициентов определить более точным способом. Такой подход обеспечивает достижение высококачественных практических результатов.

1.4. Обзор научной литературы

В статье «How does a neural network work» [2] описывается, что такое нейронные сети - это набор алгоритмов, предназначенных для распознавания закономерностей или отношений в заданном наборе данных. Они являются вычислительными системами, задуманными для имитации того, как человеческий мозг анализирует и обрабатывает информацию. Нейронные сети состоят из перцептронов, соединенных в виде сети, где каждый перцептрон выполняет математические вычисления, необходимые для классификации в соответствии с заданным набором правил.

Далее в статье рассказывается о том, как нейронные сети учатся. Процесс обучения начинается с подачи набора данных на входной слой нейронной сети, который затем передается на скрытый слой. Соединения между слоями случайным образом присваивают веса входам. Для принятия решения о состоянии активности перцептрона используется функция активации. После этого происходит сравнение выходных данных с необходимыми метками данных для вычисления окончательной ошибки. Эта ошибка используется для корректировки весов через обратное распространение (backpropagation), что позволяет уменьшить ошибку и обучить нейронную сеть. Нейронные сети легко описываются с использованием приведенной ниже диаграммы на рисунке 5. Светло-синие круги представляют

перцептроны, а линии представляют соединения между искусственными нейронами.

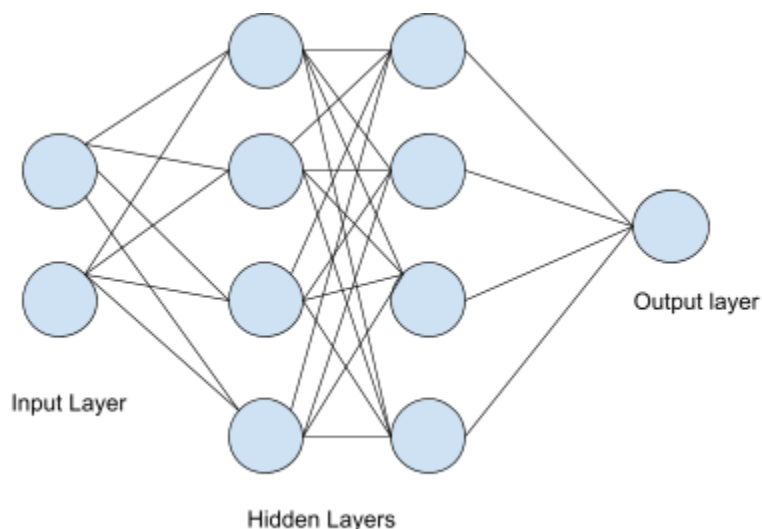


Рисунок 5 – Визуализация нейронных сетей [2]

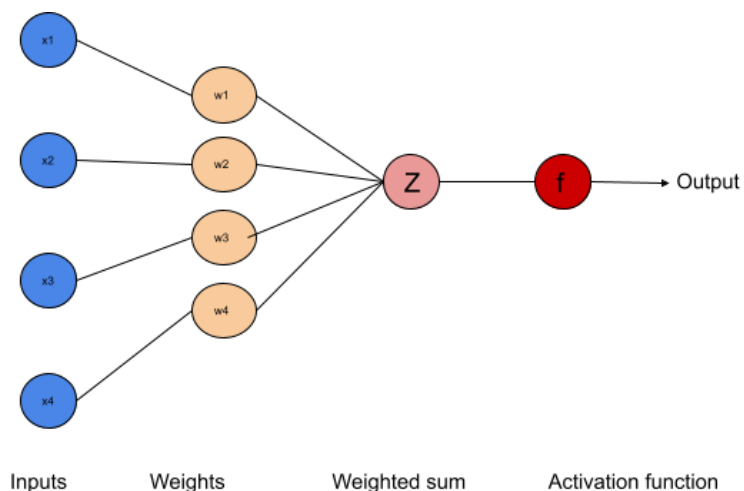


Рисунок 6 – Работа одного перцептрона [2]

На рисунке 6 рассматривается работа одного перцептрона. Его функциональность можно визуализировать следующим образом. Когда данные с произвольными весами подаются на вход модели, она генерирует их взвешенную сумму. На основании этого значения функция активации принимает решение о состоянии активации нейрона. Выход этого перцептрона может выступать в качестве входа для следующего слоя нейронов.

В статье «предсказание временных рядов с помощью Keras» [3, 11] рассматривается использование методов машинного обучения, в частности, рекуррентных нейронных сетей [15] (RNN) и полносвязных нейронных сетей (Dense), для прогнозирования временных рядов на примере курса валют.

Поставленные задачи в рассматриваемой статье:

1. Поиск данных, которые содержат информацию о курсе валюты.
2. Сбор, обработка, очистка и преобразование данных к необходимому виду, анализ данных.
3. Определение критериев оценки для исследования, поиск и выбор наиболее подходящих моделей машинного обучения, а также разработка программного обеспечения.
4. Решение задач с использованием выбранных моделей машинного обучения, проведение эмпирического исследования, сбор данных, анализ полученных результатов и их сравнение.

Показано как обрабатывались данные и какие модели нейронной сети использовались. Для прогнозирования была использована рекуррентная нейронная сеть с двумя слоями LSTM.

Рекуррентные нейронные сети [1] (РНН) представляют собой мощный инструмент для обработки последовательных данных, таких как тексты, временные ряды и аудиосигналы. Однако традиционные РНН сталкиваются с проблемой затухающего градиента, что снижает их способность улавливать долгосрочные зависимости. Для преодоления этой проблемы была разработана архитектура LSTM (Long Short-Term Memory).

LSTM имеет уникальную архитектуру, включающую три типа ворот: входные, забывания и выходные. Эти ворота контролируют поток информации внутри сети, позволяя модели решать, какую информацию сохранить, какую обновить и какую передать дальше. Такая структура помогает избежать проблемы затухания градиента, что является основной инновацией LSTM [8].

В статье также рассматривается процесс создания и обучения рекуррентной нейронной сети, а также модели полносвязной сети. На графиках были

продемонстрированы результаты обучения и проведено сравнение данных. На рисунке 7 приведены результаты прогнозирования с использованием рекуррентной нейронной сети. Синим изображена область данных, на которых обучалась модель, зеленым предсказанные значения, а оранжевым реальные значения. Как можно увидеть на графике модель хорошо справилась с предсказанием значений.

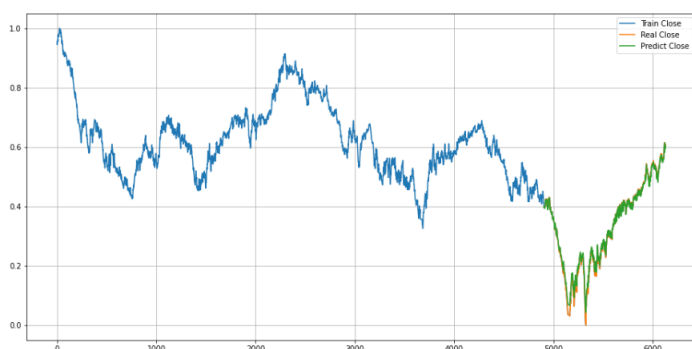


Рисунок 7 – Результаты предсказания рекуррентной нейронной сети

На рисунке 8 изображены результаты предсказания для полносвязной сети. Исходя из данных графика можно сделать вывод, что полносвязная сеть справилась с предсказанием намного хуже рекуррентной нейронной сети.

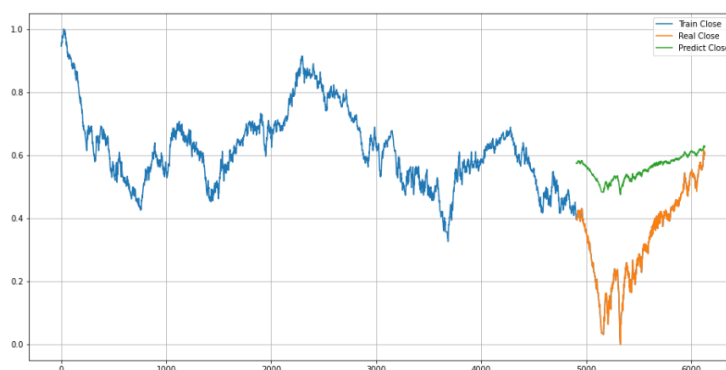


Рисунок 8 – Результаты предсказание полносвязной сети

1.5. Сравнительный анализ аналогов

В ходе исследования аналогичных программ, предназначенных для оптимизации параметров вентильных двигателей на основе их характеристик, не было обнаружено программы, которая соответствовала бы необходимым требованиям или имела бы схожий функционал.

Вывод по первой главе

В результате анализа предметной области была рассмотрена конструкция и принцип действия ВЭКВ, также была рассмотрена разработка математической схемы ВЭКВ на основе схем замещения. Был выполнен обзор литературы.

2. АНАЛИЗ ТРЕБОВАНИЙ К ПРОЕКТИРУЕМОЙ СИСТЕМЕ

2.1. Функциональные и нефункциональные требования

Проектируемая система должна соответствовать следующим функциональным требованиям.

1. Система должна считывать данные пользователя из файла формата «.csv».
2. Система должна предоставлять функционал для предварительной обработки данных, включая очистку, преобразование и масштабирование.
3. Система должна обучаться на полученных данных.
4. Система должна в результате показать графики предсказанными значениями.
5. Система должна предоставить функционал для ввода мощности ВДПТ и вывести предсказанные коэффициенты для параметров ВДПТ.

Проектируемая система должна удовлетворять следующим нефункциональным требованиям.

1. Система должна обеспечивать высокую скорость обучения и предсказания модели, чтобы пользователи могли быстро получать результаты.
2. Система должна быть реализована на языке Python.

2.2 Диаграмма вариантов использования

Для проектирования системы был использован язык графического описания UML [12] (англ. "Unified Modeling Language") – стандартизированный язык моделирования при проектировании программ. В соответствии с требованиями была построена диаграмма вариантов использования (англ. use-case diagram) – диаграмма, описывающая, какой функционал разрабатываемой программной системы доступен пользователям (рисунок 9). В нашем случае диаграмма отражает модель взаимодействия актера «Пользователь» с системой.

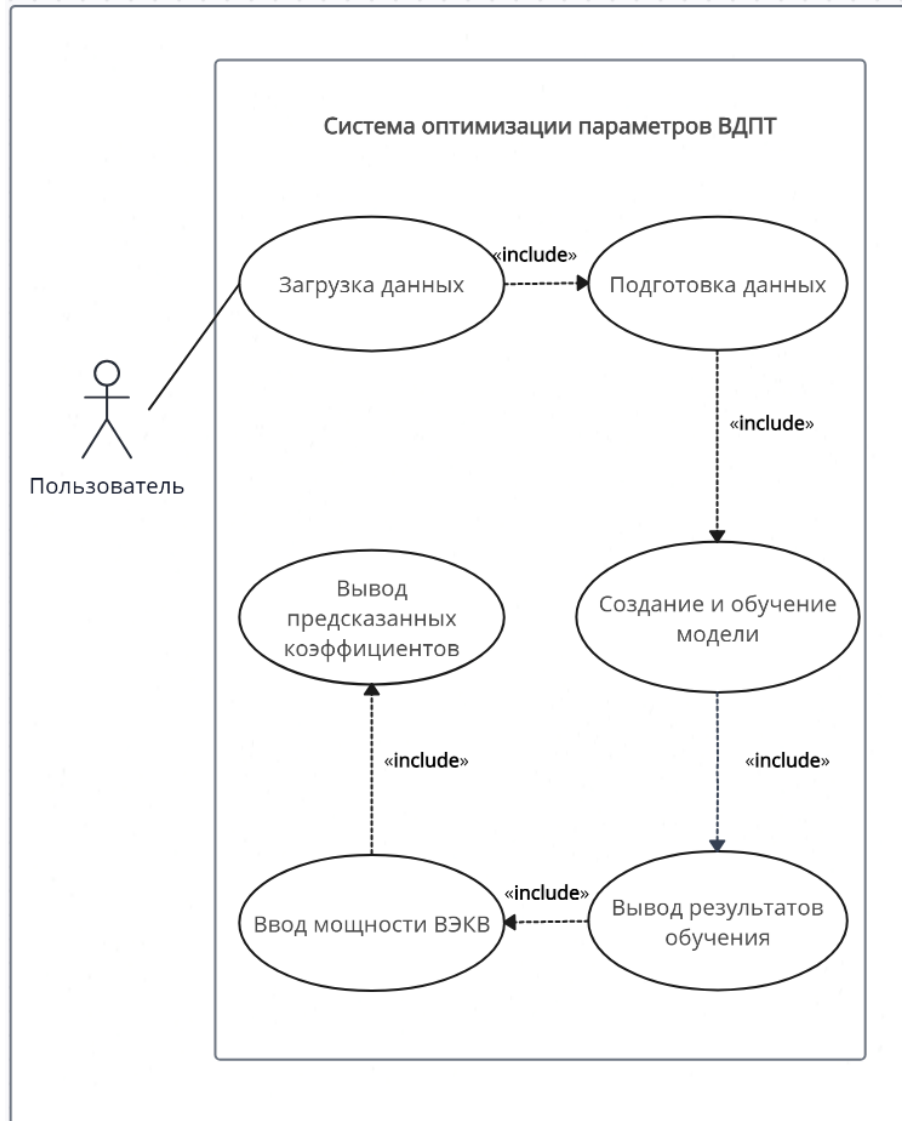


Рисунок 9 – Диаграмма вариантов использования

С системой взаимодействует только один актер – «Пользователь», использующий программную систему. Ему доступны перечисленные ниже варианты использования.

1. Вариант использования «Загрузка данных». Пользователь может загрузить набор данных из файла формата «.csv» в рабочее пространство системы.

2. Вариант использования «подготовка данных». Пользователь может запустить подготовку данных.

3. Вариант использования «Создание и обучение модели». Пользователь может запустить процесс создания и обучения модели.

4. Вариант использования «Вывод результатов обучения». Пользователь может вывести результаты обучения.

5. Вариант использования «Ввод мощности ВЭКВ». Пользователь может ввести мощность вентильного двигателя постоянного тока.

6. Вариант использования «Вывод предсказанных коэффициентов». Пользователь может вывести предсказанные коэффициенты.

Спецификация вариантов использования приведена в таблицах 1-6.

Таблица 1 – Спецификация вариантов использования «Загрузить набор данных»

Прецедент: Загрузка данных
ID: 1
Краткое описание: Загрузка исходных данных пользователем
Главные актеры: Пользователь
Второстепенные актеры: Отсутствуют
Предусловия: Отсутствуют
Основной поток: 1. Прецедент начинается, когда пользователь начинает загрузку данных. 2. Пользователь загружает в систему данные.
Постусловия: Набор данных загружен в систему
Альтернативные потоки: 1. Данные не были загружены. 2. Система выдает пользователю сообщение об ошибке.

Таблица 2 – Спецификация вариантов использования «Подготовка данных»

Прецедент: Подготовка данных
ID: 2
Краткое описание: Очистка и нормализация данных
Главные актеры: Пользователь
Второстепенные актеры: Отсутствуют
Предусловия: Данные были загружены в систему
Основной поток: 1. Прецедент начинается, когда пользователь запускает подготовку данных. 2. Система очищает данные от лишних пробелов и заменяет запятые на точки. 3. Система нормализует данные.
Постусловия: Сохраняются подготовленные данные
Альтернативные потоки: Отсутствуют

Таблица 3 – Спецификация вариантов использования «Создание и обучение модели»

Прецедент: Создание и обучение модели
ID: 3
Краткое описание: Запуск процесса обучения нейронной сети
Главные актеры: Пользователь
Второстепенные актеры: Отсутствуют
Предусловия: Данные были подготовлены
Основной поток: 4. Прецедент начинается, когда пользователь запускает обучение нейронной сети. 5. Система создает модель нейронной сети. 6. Система обучает модель нейронной сети.
Постусловия: Сохраняется обученная модель
Альтернативные потоки: Отсутствуют

Таблица 4 – Спецификация вариантов использования «Вывод результатов обучения»

Прецедент: Вывод результатов обучения
ID: 4
Краткое описание: Выводятся графики с результатами обучения модели
Главные актеры: Пользователь
Второстепенные актеры: Отсутствуют
Предусловия: В системе должна присутствовать обученная модель
Основной поток: 1. Прецедент начинается, когда пользователь запускает вывод результатов обучения. 2. Система выводит графики с результатами обучения.
Постусловия: отсутствуют
Альтернативные потоки: Отсутствуют

Таблица 5 – Спецификация вариантов использования «Ввод мощности ВЭКВ»

Прецедент: Ввод мощности ВЭКВ
ID: 5
Краткое описание: Вводится параметр мощности ВЭКВ
Главные актеры: Пользователь
Второстепенные актеры: Отсутствуют
Предусловия: В системе должна присутствовать обученная модель
Основной поток: 1. Прецедент начинается, когда пользователь запускает расчет коэффициентов. 2. Пользователь вводит мощность двигателя. 3. Система предсказывает коэффициенты с помощью обученной модели.
Постусловия: Сохраняются результаты предсказания
Альтернативные потоки: Отсутствуют

Таблица 6 – Спецификация вариантов использования «Вывод предсказанных коэффициентов»

Прецедент: Вывод предсказанных коэффициентов
ID: 6
Краткое описание: Выводятся числовые значения предсказанных коэффициентов
Главные актеры: Пользователь
Второстепенные актеры: Отсутствуют
Предусловия: В системе должны присутствовать предсказанные коэффициенты
Основной поток: 1. Прецедент начинается, когда пользователь запускает вывод информации предсказанных коэффициентов. 2. Система выводит числовые значения предсказанных коэффициентов.
Постусловия: Отсутствуют
Альтернативные потоки: Отсутствуют

Вывод по второй главе

Во второй главе были определены функциональные и нефункциональные требования к системе. На основе этих требований была построена UML диаграмма вариантов использования, определены основные актеры - пользователь, взаимодействующие с системой, а также приведено краткое описание всех вариантов использования и спецификации вариантов использования, где были описаны все сценарии использования системы и детали, связанные с их исполнением.

3. РЕАЛИЗАЦИЯ

3.1. Программные средства реализации

Для реализации модели нейронной сети, сбора и предобработки данных выборки, и обучения нейронных сетей был использован язык программирования Python 3.9. Для написания кода и проведения экспериментов была использована облачная вычислительная среда Google Colab [10].

В процессе разработки были использованы следующие библиотеки и фреймворки.

1. TensorFlow (2.16.1) [4]. открытая программная библиотека для машинного обучения, разработанная компанией Google. Она предоставляет инструменты для построения и обучения различных моделей машинного обучения, включая нейронные сети.

2. Sklearn (0.20). [5]. Библиотека для машинного обучения в Python. Она предоставляет простой и эффективный способ решения задачи масштабирования данных.

3. Numpy (1.19.2) [6]. Библиотека, для работы с матрицами и многомерными массивами.

4. Pandas (2.2.2) [7]. Это инструмент для анализа и обработки данных в языке программирования Python. Pandas предоставляет высокоуровневые структуры данных и функции, которые облегчают выполнение множества операций над данными, таких как фильтрация, агрегация, сортировка, объединение и многое другое.

5. Matplotlib (3.9.0) [8]. Библиотека для построения различных типов графиков и диаграмм.

6. SciPy (1.13.1). [9]. Библиотека для научных и инженерных вычислений в языке программирования Python. Она построена на базе библиотеки NumPy и предоставляет множество функций для выполнения сложных математических операций, анализа данных и решения различных задач в науке и технике.

3.2. Подготовка данных

Анализ характеристик электрических машин демонстрирует, что их энергоэффективность и удельные энергетические показатели значительно зависят от соотношений площадей активных зон.

Эти параметры рассматриваются как обобщенные переменные и имеют четкие границы изменения от 0 до 1. Они указывают на оптимальное соотношение активных зон, участвующих в преобразовании энергии, и их оптимальные значения остаются в узком диапазоне, несмотря на изменения габаритов в широких пределах.

1. Переменная f_s – показывает то, какую часть от площади поперечного сечения электрической машины занимают активные части электрической машины (рисунок 10).

$$f_s = \frac{S_{\text{кольца}}}{S_{\text{круга}}} = \frac{S_{\text{поп.сеч.маш}}}{S_{\text{поп.сеч}}}, \quad (1)$$

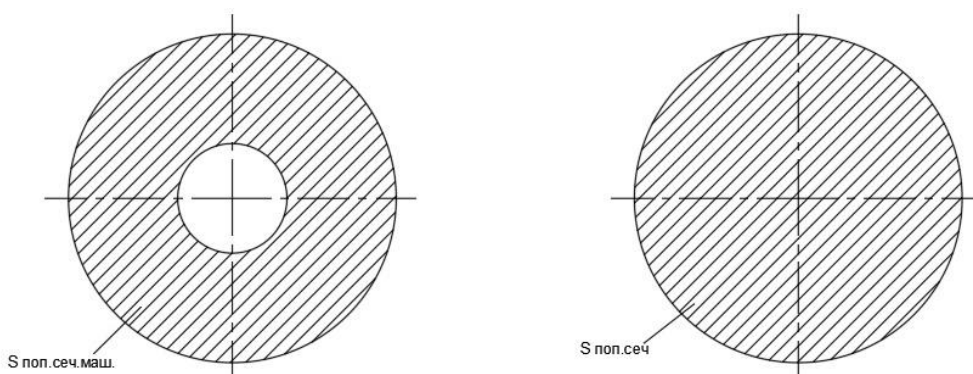


Рисунок 10 – Переменная f_s

2. Переменная f_e – отражает отношение площади поверхности, занимаемой ротором, к площади поверхности, которую занимают активные части машины (рисунок 11).

$$f_e = \frac{S_{\text{ротора}}}{S_{\text{поп.сеч.маш}}}, \quad (2)$$

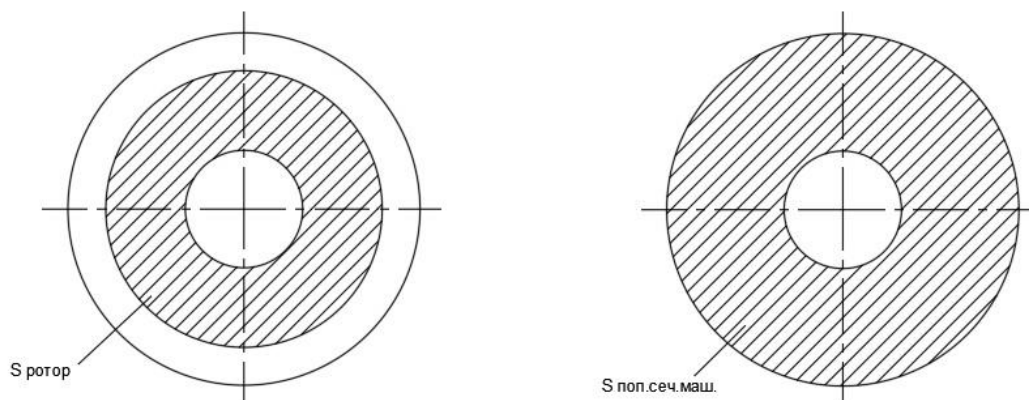


Рисунок 11 – Переменная f_e

3. Переменная f_{ov} – показывает отношение площади поверхности пазово-зубцового слоя ротора к площади поверхности, которую занимает сам ротор (рисунок 12).

$$f_{ov} = \frac{S_{\text{паз.зубц.слой ротора}}}{S_{\text{ротора}}}, \quad (3)$$

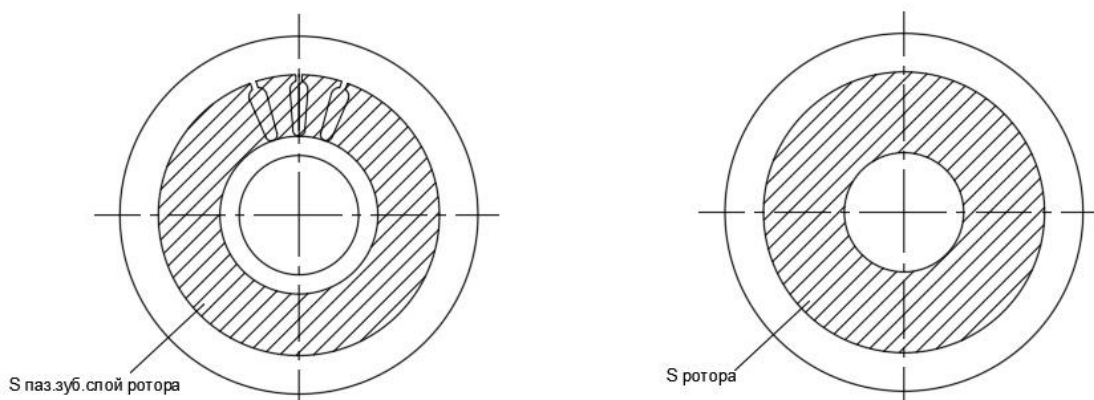


Рисунок 12 – Переменная f_{ov}

4. Переменная f_j – отражает отношение площади поверхности, занимаемой пазами ротора, к площади поверхности, которую занимает пазово-зубцовый слой ротора (рисунок 13).

$$f_j = \frac{S_{\text{пазов ротора}}}{S_{\text{паз.зубц.слой ротора}}}, \quad (4)$$

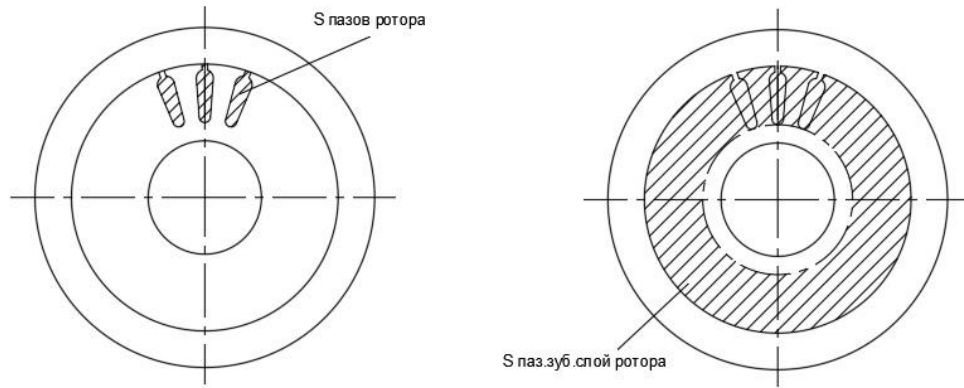


Рисунок 13 – Переменная f_j

5. Переменная f_m – показывает отношение площади поверхности пазово-зубцового слоя статора к площади поверхности, которую занимает сам статор (рисунок 14).

$$f_m = \frac{S_{\text{паз.зубц.слоя статора}}}{S_{\text{статора}}}, \quad (5)$$

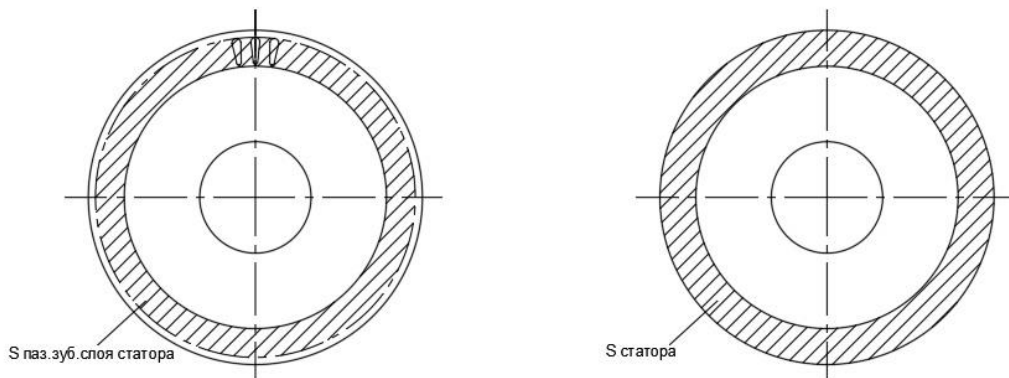


Рисунок 14 – Переменная f_m

6. Переменная f_{zm} – отражает отношение площади поверхности, занимаемой пазами статора, к площади поверхности, которую занимает пазово-зубцовый слой статора (рисунок 15).

$$f_{zm} = \frac{S_{\text{пазов статора}}}{S_{\text{паз.зубц.слоя статора}}}, \quad (6)$$

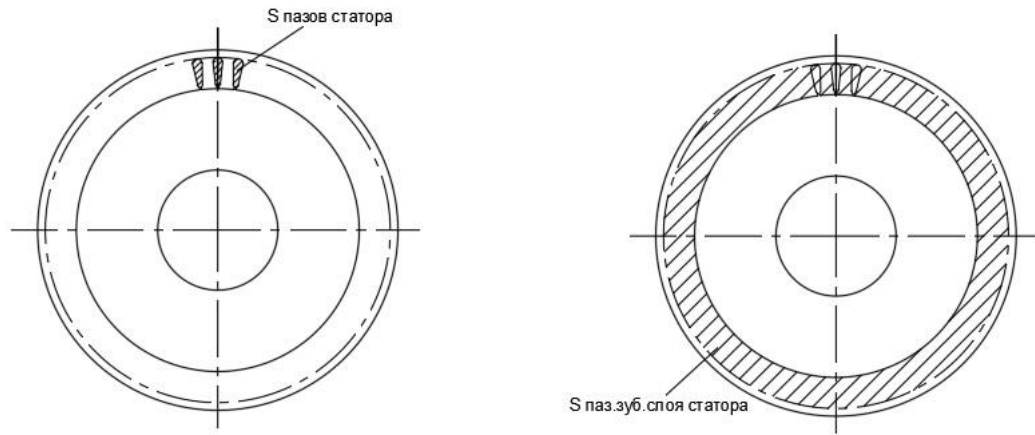


Рисунок 15 – Переменная f_{zm}

В ходе разработки инженерной методики, были построены следующие графики, значения которых на прямую влияют на геометрию, расчет электромашин и дает возможность решить задачи однокритериальной многоуровневой оптимизации.

На рисунке 16 представлен график для первого параметра «Минимальная масса активных материалов».

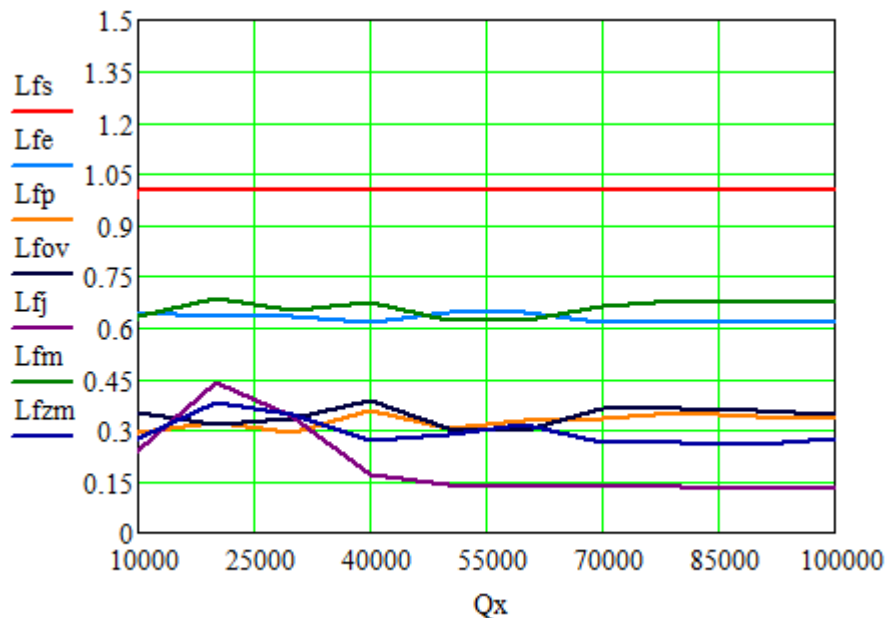


Рисунок 16 – Минимальная масса активных материалов

В таблице 7 представлены исходные данные обобщенных переменных для графика минимальной массы активных материалов.

Таблица 7 – данные графика «Минимальной массы активных материалов»

Qx	Lfs	Lfe	Lfp	Lfov	Lfj	Lfm	Lfzm
10000,000	1,000	0,646	0,294	0,352	0,242	0,632	0,275
20000,000	1,000	0,630	0,321	0,317	0,440	0,683	0,378
30000,000	1,000	0,630	0,293	0,336	0,340	0,652	0,348
40000,000	1,000	0,614	0,355	0,384	0,170	0,675	0,271
50000,000	1,000	0,646	0,304	0,307	0,143	0,624	0,290
60000,000	1,000	0,646	0,328	0,297	0,135	0,623	0,314
70000,000	1,000	0,614	0,355	0,366	0,143	0,663	0,266
80000,000	1,000	0,614	0,350	0,362	0,134	0,679	0,264
90000,000	1,000	0,614	0,338	0,355	0,129	0,673	0,259
100000,000	1,000	0,613	0,336	0,347	0,130	0,670	0,278

На рисунке 17 представлен график для второго параметра «Минимальная масса магнитов».

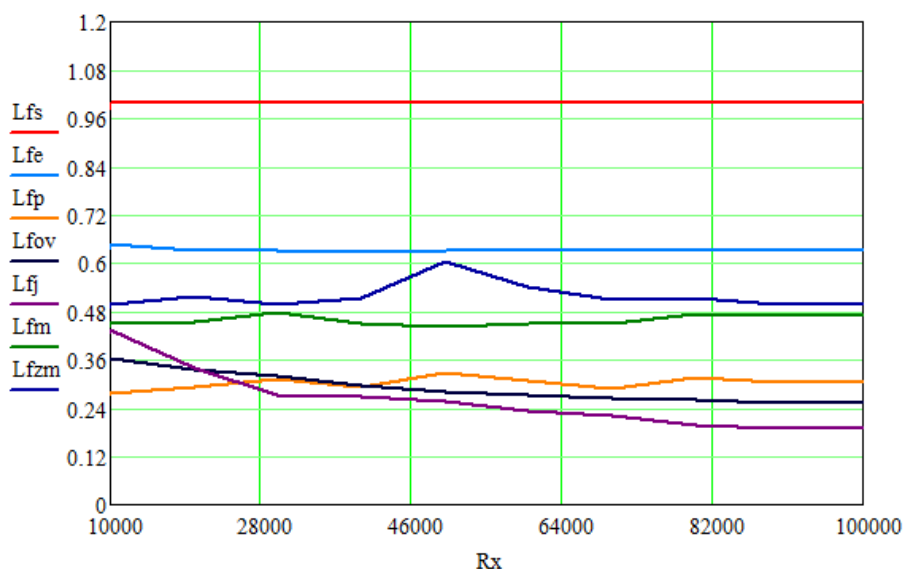


Рисунок 17 – Минимальная масса магнитов

В таблице 8 представлены исходные данные обобщенных переменных к графику «Минимальная масса магнитов».

Таблица 8 – Данные к графику «Минимальная масса магнитов»

Rx	Lfs	Lfe	Lfp	Lfov	Lfj	Lfm	Lfzm
10000,000	1,000	0,630	0,278	0,364	0,435	0,450	0,496
20000,000	1,000	0,630	0,293	0,336	0,340	0,452	0,518
30000,000	1,000	0,630	0,310	0,321	0,271	0,478	0,497
40000,000	1,000	0,629	0,291	0,297	0,268	0,451	0,514
50000,000	1,000	0,630	0,329	0,281	0,259	0,441	0,602
60000,000	1,000	0,630	0,306	0,272	0,234	0,450	0,540
70000,000	1,000	0,629	0,290	0,263	0,222	0,451	0,510
80000,000	1,000	0,630	0,315	0,261	0,197	0,473	0,515
90000,000	1,000	0,630	0,304	0,253	0,190	0,471	0,498
100000,000	1,000	0,630	0,304	0,253	0,190	0,471	0,498

На рисунке 18 представлен график для третьего параметра «Минимальный объем».

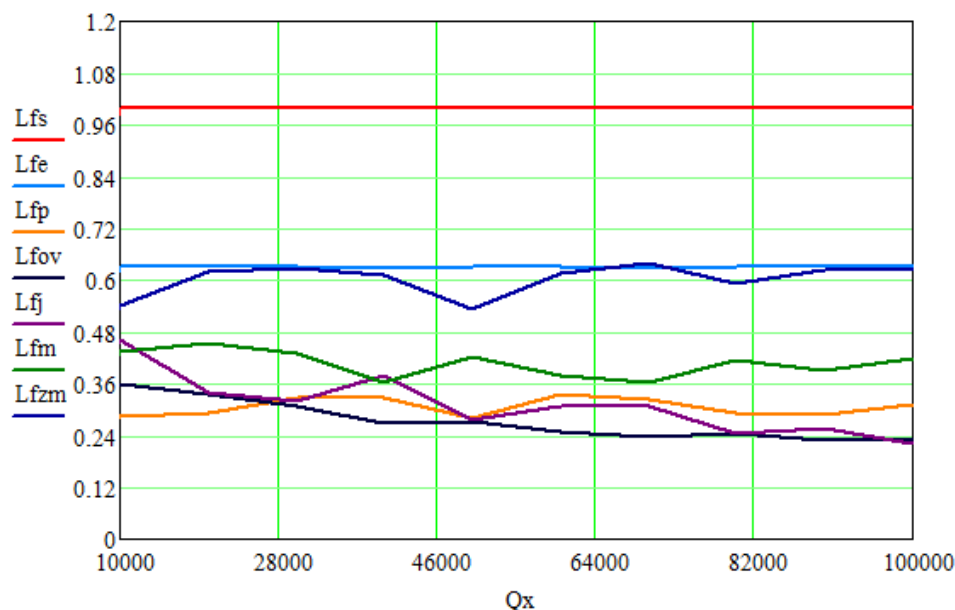


Рисунок 18 – Минимальный объем

В таблице 9 представлены исходные данные обобщенных переменных к графику «Минимальный объем»

Таблица 9 – данные к графику «Минимальный объем»

Qx	Lfs	Lfe	Lfp	Lfov	Lfj	Lfm	Lfzm
10000,000	1,000	0,630	0,284	0,361	0,462	0,434	0,539
20000,000	1,000	0,630	0,293	0,336	0,340	0,452	0,618
30000,000	1,000	0,630	0,329	0,308	0,318	0,432	0,626
40000,000	1,000	0,629	0,327	0,268	0,377	0,363	0,613
50000,000	1,000	0,630	0,281	0,274	0,277	0,421	0,533
60000,000	1,000	0,629	0,335	0,250	0,306	0,380	0,616
70000,000	1,000	0,630	0,325	0,235	0,308	0,365	0,639
80000,000	1,000	0,630	0,291	0,243	0,243	0,413	0,591
90000,000	1,000	0,630	0,289	0,229	0,256	0,389	0,623
100000,000	1,000	0,630	0,313	0,232	0,220	0,418	0,622

Поскольку, во всех трех параметрах коэффициент Lfs является неизменным, его можно убрать из набора данных, чтобы избежать переобучения нейронной сети.

Для эффективного обучения модели необходимо увеличить представленный набор исходных данных. Вот несколько причин, почему большой набор данных необходим:

- обобщающая способность модели. Обобщение – это способность модели хорошо работать на новых, ранее не виденных данных. Большие наборы данных

помогают модели учиться на различных образцах, что улучшает её способность обобщать и снижает вероятность переобучения;

- разнообразие данных. Разнообразие в данных позволяет модели изучать все возможные вариации входных данных;

- уменьшение смещения. Малые наборы данных могут охватывать не все вариации, что приводит к смещению модели в сторону определенных паттернов. Большие наборы данных уменьшают вероятность смещения и помогают модели более точно отображать реальное распределение данных;

- избежание переобучения. Переобучение происходит, когда модель слишком хорошо запоминает обучающие данные и плохо работает на новых данных;

- улучшение точности модели. Чем больше данных, тем лучше модель может изучать и распознавать сложные паттерны и зависимости;

- проверка и оценка модели. Большие наборы данных позволяют разделить данные на обучающую, валидационную и тестовую выборки, что важно для объективной оценки модели.

Для увеличения набора данных был использован метод линейной интерполяции. Линейная интерполяция предполагает, что между каждой парой точек данные изменяются линейно. То есть, для каждой новой точки, ее значение рассчитывается как взвешенная сумма ближайших известных значений. Код для интерполяции данных изображен в листинге 1.

Листинг 1 – Интерполяция исходных данных

```
#Чтение данных из файла
data = pd.read_csv('params.csv')

# Очистка данных: убираем пробелы и заменяем запятые на точки
data = data.apply(lambda x: x.str.replace('\s+', '', regex=True) if
x.dtype == "object" else x)
for column in data.columns:
    if data[column].dtype == 'object':
        data[column] = data[column].str.replace(',', ',
'.').astype(float)
# Разделим данные на значения x и соответствующие значения Lfs, Lfe,
Lfp, Lfov, Lfj, Lfm, Lfzm
```



```

X_original = data['x'].values
Y_original = data[['Lfs', 'Lfe', 'Lfp', 'Lfov', 'Lfj', 'Lfm',
'Lfzm']].values

# Создаем новые значения x с шагом в 100 между существующими точками
new_X = np.arange(X_original[0], X_original[-1] + 100, 100)

# Интерполируем значения Lfs, Lfe, Lfp, Lfov, Lfj, Lfm, Lfzm для
новых значений x
new_Y = np.zeros((len(new_X), Y_original.shape[1]))
for i in range(Y_original.shape[1]):
    interp_function = interp1d(X_original, Y_original[:, i],
kind='linear')
    new_Y[:, i] = interp_function(new_X)

# Создаем новый DataFrame с интерполированными данными
new_data = pd.DataFrame({
    'x': new_X,
    'Lfs': new_Y[:, 0],
    'Lfe': new_Y[:, 1],
    'Lfp': new_Y[:, 2],
    'Lfov': new_Y[:, 3],
    'Lfj': new_Y[:, 4],
    'Lfm': new_Y[:, 5],
    'Lfzm': new_Y[:, 6]
})

# Сохраняем новые данные в CSV файл
new_data.to_csv('interpolated_params.csv', index=False)

```

Сначала происходит чтение данных из файла. После чего происходит очистка данных (убираются лишние пробелы и заменяются запятые на точки), разделение данных на «X» и «Y» - входные и выходные данные соответственно. Следующим шагом создается массив для новых значений X с шагом 100 между существующими точками X в исходных данных. Далее создается массив для новых значений Y. Для каждой колонки в «Y_original» создается функция интерполяции «interp_function» с использованием функции «interp1d» из библиотеки SciPy. В конце новые данные записываются в датафрейм и сохраняются в файл.

3.3. Реализация модели многослойного перцептрона

Для поиска лучших параметров был использован метод Grid Search. Grid search – это метод автоматического поиска лучших гиперпараметров для модели машинного обучения. Далее представлен код.

Листинг 2 – Обработка и подготовка данных

```
# Чтение данных из файла
data = pd.read_csv('interpolated_params.csv')

# Очистка данных: убираем пробелы и заменяем запятые на точки
data = data.apply(lambda x: x.str.replace('\s+', '', regex=True) if
x.dtype == "object" else x)
for column in data.columns:
    if data[column].dtype == 'object':
        data[column] = data[column].str.replace(',',
'.').astype(float)

# Извлекаем X и Y после очистки данных
X = data[['x']].values
Y = data[['Lfe', 'Lfp', 'Lfov', 'Lfj', 'Lfm', 'Lfzm']].values

# Нормализация данных
scaler_x = StandardScaler()
scaler_y = StandardScaler()
X = scaler_x.fit_transform(X)
Y = scaler_y.fit_transform(Y)

# Разделение данных на обучающую и тестовую выборки
X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
test_size=0.2, random_state=42)
```

В листинге 2 происходит чтение интерполированных данных из файла, очистка данных от пробелов и замена запятых на точки. В переменную X добавляются мощность двигателя, а в переменную Y все коэффициенты. После чего происходит нормализация данных и разделение на обучающую и тестовую выборки. Обучающая выборка 80% и тестовая 20%.

В листинге 3 функция создания модели. В ней создается модель и последовательно добавляются слои:

1. Первый слой: Dense(64, input_shape=(1,), activation=activation). Полносвязный слой с 64 нейронами. Входной слой, который принимает один входной признак – мощность двигателя.

2. Второй слой: Dense(64, activation=activation). Полносвязный слой с 64 нейронами.

3. Третий слой: Dense(6). Полносвязный слой с 6 нейронами.

Листинг 3 – создание модели

```
# Создание модели многослойного перцептрона
def create_model(learning_rate=0.001, activation='relu'):
    model = Sequential()
    model.add(Dense(64, input_shape=(1,), activation=activation))
    model.add(Dense(64, activation=activation))
    model.add(Dense(6))
    model.compile(optimizer=Adam(learning_rate=learning_rate),
loss='mean_squared_error')
    return model
```

Для компиляции модели используется оптимизатор «Adam» (Adaptive Moment Estimation) – это алгоритм оптимизации, который сочетает в себе преимущества двух других популярных оптимизаторов: «AdaGrad» и «RMSProp». Данный оптимизатор обеспечивает автоматическую настройку скорости обучения и момента для эффективной оптимизации модели.

В качестве функции потерь (loss function) используется «mean_squared_error» (среднеквадратичная ошибка). MSE вычисляется как среднее арифметическое квадратов разностей между фактическими и предсказанными значениями для всех обучающих примеров.

Листинг 4 – Grid Search

```
# Определение пространства гиперпараметров для поиска
param_grid = {
    'batch_size': [2, 4, 8, 16],
    'epochs': [50, 75, 100],
    'learning_rate': [0.001, 0.01, 0.1],
    'activation': ['relu', 'tanh']
}
# Функция для создания модели с параметрами
def build_model(learning_rate, activation):
    return KerasRegressor(model=create_model,
learning_rate=learning_rate, activation=activation, verbose=0)
# Настройка Grid Search
grid =
GridSearchCV(estimator=build_model(learning_rate=learning_rate,
activation=activation), param_grid=param_grid, n_jobs=-1, cv=3)
grid_result = grid.fit(X_train, Y_train)
print(f"Лучшие параметры: {grid_result.best_params}")
```

В листинге 4 в param_grid задаются диапазоны значений для гиперпараметров. Build_model – функция для создания модели. Настраиваем Grid

Search и подаем входные и выходные данные. После чего выводим лучшие параметры.

В листинге 5 код для класса, который определяет пользовательский коллбэк TimeHistory, который расширяет базовый класс Callback из Keras. Этот коллбэк позволяет измерять время, затраченное на каждую эпоху обучения модели.

Листинг 5 – Класс для измерения времени обучения

```
# Коллбек для измерения времени обучения
class TimeHistory(Callback):
    def on_train_begin(self, logs={}):
        self.times = []

    def on_epoch_begin(self, epoch, logs={}):
        self.epoch_time_start = time.time()

    def on_epoch_end(self, epoch, logs={}):
        self.times.append(time.time() - self.epoch_time_start)

time_callback = TimeHistory()
```

Он работает следующим образом:

1) в методе on_train_begin коллбэк инициализирует пустой список times, в который будут записываться временные показатели для каждой эпохи;

2) в методе on_epoch_begin коллбэк сохраняет текущее время перед началом каждой эпохи;

3) в методе on_epoch_end коллбэк рассчитывает время, прошедшее с начала эпохи до ее завершения, путем вычитания времени начала эпохи из текущего времени. Затем это время добавляется в список times.

В листинге 7 код, который отвечает за обучение модели с лучшими параметрами.

Листинг 7 – обучение модели

```
# Оценка модели с лучшими параметрами
best_model =
create_model(learning_rate=grid_result.best_params_['learning_rate']
, activation=grid_result.best_params_['activation'])
history = best_model.fit(X_train, Y_train,
epochs=grid_result.best_params_['epochs'],
batch_size=grid_result.best_params_['batch_size'],
validation_split=0.2, verbose=1, callbacks=[time_callback])
```

```

# Оценка модели на тестовых данных
loss, mae = best_model.evaluate(X_test, Y_test, verbose=1)
print(f'Test Loss: {loss}')
print(f'Test MAE: {mae}')

# Вывод времени обучения
print(f"Время обучения: {sum(time_callback.times)}")

```

Метрика «MAE» (средняя абсолютная ошибка) используется для оценки производительности модели. Она показывает среднюю абсолютную разницу между предсказанными и истинными значениями.

В результате работы алгоритма для разного количества нейронов лучшая функция активации во всех случаях RELU, лучший показатель learning_rate во всех случаях равен 0.001, количество эпох равно 100, также для любого количество нейронов. Остальные результаты работы алгоритма приведены в таблице 9.

Таблица 9 – результаты работы

Количество нейронов	Размер пакета	Потери	MAE	Время обучения в секундах
32	2	0,00029	0,01273	87
64	4	0,00030	0,01320	45
128	2	0,00041	0,01430	97

На этих данных можно увидеть, что потери и абсолютная разница несущественно отличаются для любого количества нейронов, однако скорость обучения при использовании 64 нейронов и размера пакета 4 существенно выше. Поэтому для создания нейронной сети многослойного перцептрона были определены следующие показатели:

1. Входной слой – признаки передаются в нейроны входного слоя. В данном случае, размерность входного слоя равна 1, что соответствует количеству признаков в обучающих данных.

2. Скрытый слой 1 – В данном слое используется 64 нейрона с функцией активации ReLU (Rectified Linear Unit). Функция активации ReLU применяется для введения нелинейности в модель. Каждый нейрон в этом слое принимает входные данные и вычисляет взвешенную сумму, а затем применяет функцию активации ReLU к полученному результату.

3. Скрытый слой 2 – в данном слое используется 64 нейрона с функцией активации ReLU. Подобно первому скрытому слою, функция активации ReLU используется для введения нелинейности.

4. Выходной слой – в данном случае, используется 6 нейронов, соответствующих количеству выходных значений. Линейная функция активации.

Графическое представление архитектуры представлено на рисунке 19.

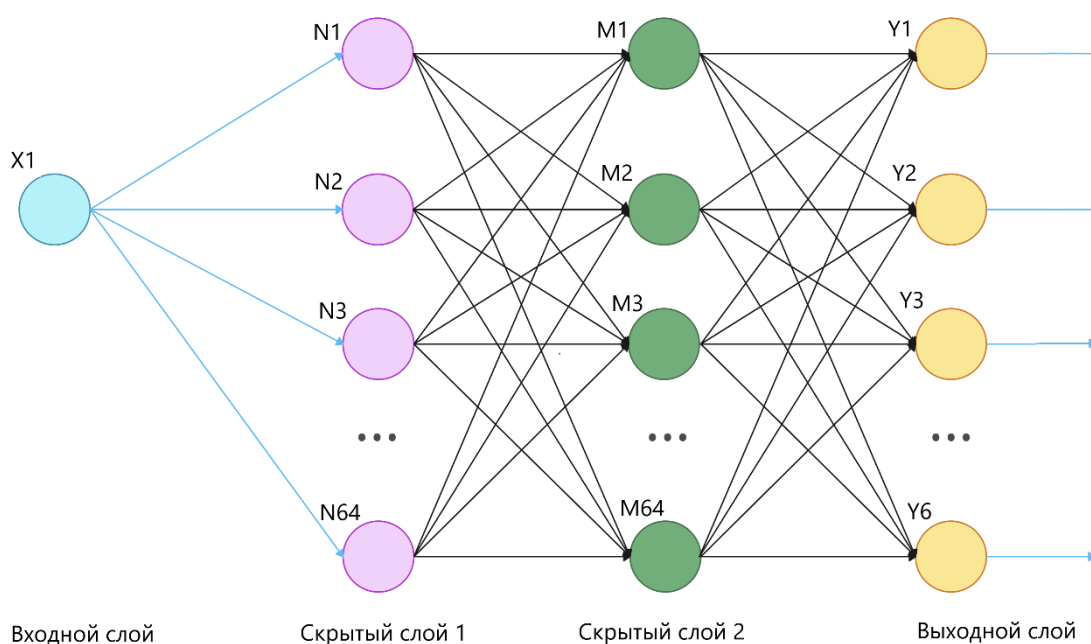


Рисунок 19 – Архитектура нейронной сети

В листинге 8 приведен финальный вариант модели многослойного перцептрона с лучшими параметрами.

Листинг 8 – модель многослойного перцептрона

```
model = Sequential()

# Добавление полносвязных слоев
model.add(Dense(64, activation='relu', input_shape=(1,)))
model.add(Dense(64, activation='relu'))
model.add(Dense(6)) # 6 выходных нейронов для предсказания

# Компиляция модели
model.compile(optimizer=Adam(learning_rate=0.001),
              loss='mean_squared_error', metrics=['mae'])
```

```
# Обучение модели
history = model.fit(X_train, Y_train, epochs=100, batch_size=4,
validation_split=0.2, verbose=1,
callbacks=[time_callback])
```

3.4. Реализация модели рекуррентной нейронной сети

Для начала необходимо найти оптимальные гиперпараметры для разрабатываемой нейронной сети. Для поиска оптимальных гиперпараметров используется GridSearchCV с различными значениями batch_size, epochs, learning_rate и функцией активации.

Лучшая модель обучается с выбранными гиперпараметрами, и производительность оценивается на тестовых данных. Используется коллбек для измерения времени обучения.

Листинг 9 – Чтение и очистка данных

```
data = pd.read_csv('interpolated_params.csv')

# Очистка данных: убираем пробелы и заменяем запятые на точки
data = data.apply(lambda x: x.str.replace('\s+', '', regex=True) if
x.dtype == "object" else x)
for column in data.columns:
    if data[column].dtype == 'object':
        data[column] = data[column].str.replace(',',
'.').astype(float)
```

Листинг 10 – Разделение на X и Y и нормализация данных

```
X_original = data['x'].values
Y_original = data[['Lfe', 'Lfp', 'Lfov', 'Lfj', 'Lfm',
'Lfzm']].values

# Нормализация данных
scaler_x = StandardScaler()
scaler_y = StandardScaler()
X = scaler_x.fit_transform(X_original.reshape(-1, 1))
Y = scaler_y.fit_transform(Y_original)
```

Листинг 11 – Преобразование данных, разделение на обучающую и тестовую выборки

```
# Преобразование данных для RNN
time_steps = 1
X_rnn = np.reshape(X, (X.shape[0], time_steps, 1))

# Разделение данных на обучающую и тестовую выборки
```

```
X_train, X_test, Y_train, Y_test = train_test_split(X_rnn, Y,
test_size=0.2, random_state=42)
```

Для LSTM слой требует ввода данных в форме (samples, time_steps, features:

- 1) параметр samples – это входные данные;
- 2) параметр time_steps задает количество временных шагов (временной контекст), которые LSTM будет обрабатывать за один раз;
- 3) параметр features соответствует количеству признаков в каждом временном шаге.

Листинг 12 – Функция создания модели

```
def create_rnn_model(learning_rate=0.001, activation='relu'):
    model = Sequential()
    model.add(LSTM(128, activation=activation,
input_shape=(time_steps, 1)))
    model.add(Dense(128, activation=activation))
    model.add(Dense(6))
    model.compile(optimizer=Adam(learning_rate=learning_rate),
loss='mean_squared_error')
    return model
```

Листинг 13 – Определение пространства гиперпараметров для поиска

```
param_grid = {
    'batch_size': [2, 4, 8, 16],
    'epochs': [50, 75, 100],
    'learning_rate': [0.001, 0.01],
    'activation': ['relu', 'tanh']
}
```

Листинг 14 – Настройка Grid search и вывод лучших параметров

```
def build_model(learning_rate, activation):
    return KerasRegressor(model=create_rnn_model,
learning_rate=learning_rate, activation=activation, verbose=0)
# Настройка Grid Search
grid = GridSearchCV(estimator=build_model(learning_rate=0.001,
activation='relu'), param_grid=param_grid, n_jobs=-1, cv=3)
grid_result = grid.fit(X_train, Y_train)
# Вывод лучших параметров
print(f"Лучшие параметры: {grid_result.best_params}")
```


Листинг 15 – Оценка модели

```
# Оценка модели с лучшими параметрами
best_model =
create_rnn_model(learning_rate=grid_result.best_params_['learning_rate'],
activation=grid_result.best_params_['activation'])
history = best_model.fit(X_train, Y_train,
epochs=grid_result.best_params_['epochs'],
batch_size=grid_result.best_params_['batch_size'],
validation_split=0.2, verbose=1, callbacks=[time_callback])
# Оценка модели на тестовых данных
loss, mae = best_model.evaluate(X_test, Y_test, verbose=1)
print(f'Test Loss: {loss}')
print(f'Test MAE: {mae}')
# Вывод времени обучения
print(f"Время обучения на каждой эпохе: {sum(time_callback.times)}")
```

В результате работы алгоритма для разного количества нейронов лучшая функция активации во всех случаях RELU, лучший показатель learning_rate во всех случаях равен 0.001, количество эпох равно 100, также для любого количество нейронов. Остальные результаты работы алгоритма приведены в таблице 10.

Таблица 10 – результаты работы

Количество нейронов	Размер пакета	Потери	MAE	Время обучения в секундах
32	2	0,00030	0,01501	133
64	4	0,00060	0,01430	73
128	8	0,00027	0,01400	46

На этих данных можно увидеть, что по всем показателям вариация с 128 нейронами и размером пакета 8 превосходит остальные варианты. Поэтому для создания нейронной сети были определены следующие показатели:

1. Входной слой – признаки передаются в нейроны входного слоя. В данном случае размерность входного слоя равна (1, 1), что соответствует одному временному шагу и одному признаку в обучающих данных.

2. LSTM слой (скрытый слой 1) – В данном слое используется 128. LSTM слой способен запоминать долгосрочные зависимости и обрабатывать последовательные данные. Каждый нейрон принимает входные данные, вычисляет взвешенную сумму и передает состояние следующему временному шагу. Функция

активации ReLU (Rectified Linear Unit) применяется для введения нелинейности в модель.

3. Полносвязный скрытый слой (скрытый слой 2) – В данном слое используется 128 нейронов с функцией активации ReLU. Подобно первому скрытому слою, функция активации ReLU используется для введения нелинейности, что позволяет модели обучаться более сложным зависимостям.

4. Выходной слой – состоит из 6 нейронов, что соответствует количеству выходных значений. В данном случае используется линейная функция активации, так как задача регрессии требует предсказания непрерывных значений.

Компиляция модели: оптимизатор Adam используется для адаптивной настройки скорости обучения; в качестве функции потерь используется `mean_squared_error`, что подходит для задач регрессии; также в качестве метрики используется MAE (Mean Absolute Error) для оценки производительности модели.

Графическое представление архитектуры представлено на рисунке 20.

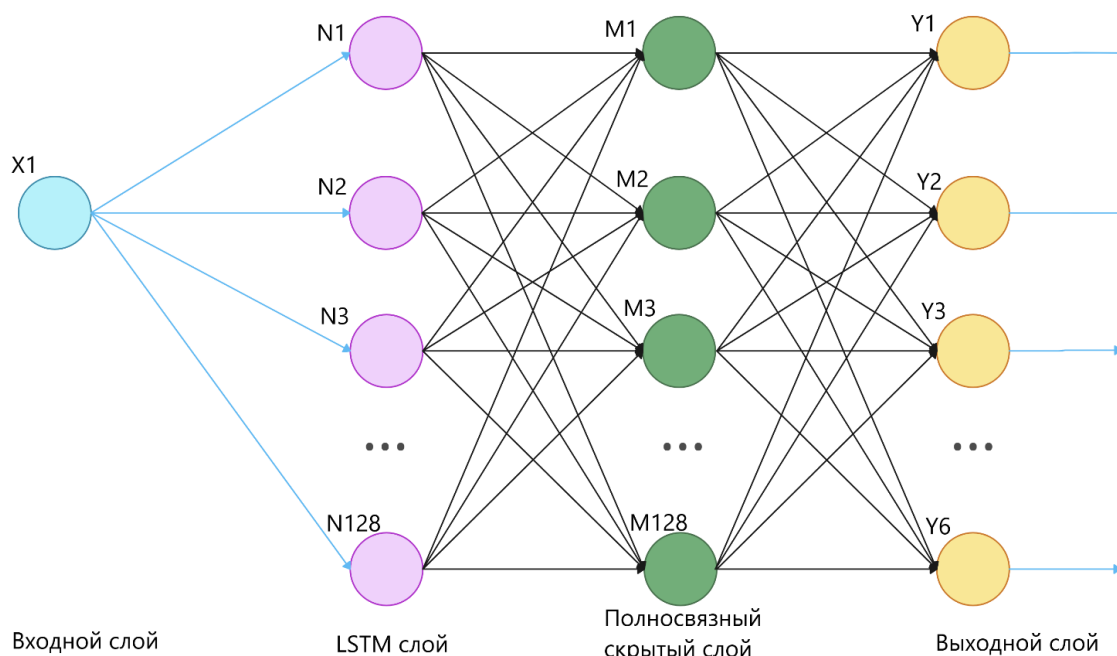


Рисунок 20 – Архитектура рекуррентной нейронной сети

В листинге 16 приведен финальный вариант модели рекуррентной нейронной сети с лучшими параметрами.

Листинг 16 – Модель рекуррентной нейронной сети

```
# Создание модели RNN
model = Sequential()
model.add(LSTM(128, activation='relu', input_shape=(time_steps, 1)))
model.add(Dense(128, activation='relu'))
model.add(Dense(6)) # 6 выходных нейронов для предсказания

learning_rate = 0.001
optimizer = Adam(learning_rate=learning_rate)

# Компиляция модели
model.compile(optimizer=optimizer, loss='mean_squared_error',
metrics=['mae'])

# Вывод информации о модели
model.summary()

# Обучение модели
history = model.fit(X_train, Y_train, epochs=100, batch_size=8,
validation_split=0.2, verbose=1, callbacks=[time_callback])
```

3.5. Реализация функций предсказания и построения графиков

В листинге 17 приведен код для функции предсказания коэффициентов на основе введенного значения мощности двигателя.

Листинг 17 – функция для предсказания

```
# Функция для предсказания на основе введенного значения X
def predict_y(input_x):
    input_x = scaler_x.transform(np.array(input_x).reshape(-1, 1))
# Нормализуем входные данные
    prediction = model.predict(input_x)
    prediction = scaler_y.inverse_transform(prediction) # Де-
нормализуем выходные данные
    return prediction

input_x = float(input('Введите мощность двигателя: '))
predicted_y = predict_y(input_x)

# Список названий для каждого элемента в predicted_y
output_names = ['Lfe', 'Lfp', 'Lfov', 'Lfj', 'Lfm', 'Lfzm']

# Вывод значений с их названиями
print(f'Input: {input_x}')
```

```
print('Predicted Output:')
for name, value in zip(output_names, predicted_y[0]):
    print(f'{name}: {value}')
```

Функция работает следующим образом: `input_x` преобразуется в массив NumPy и изменяет форму на $(-1, 1)$, чтобы соответствовать ожидаемому формату; `scaler_x` использует `StandardScaler`, обученный на обучающем наборе данных, чтобы нормализовать входные данные. Нормализация приводит данные к стандартному виду (среднее значение 0, стандартное отклонение 1); нормализованные входные данные передаются в модель для предсказания. Модель возвращает нормализованные предсказанные значения; `scaler_y` использует `StandardScaler`, обученный на выходных данных, чтобы де-нормализовать предсказанные значения. Это приводит предсказанные значения к их исходному масштабу; функция возвращает де-нормализованные предсказанные значения.

Функция для построения графиков для сравнения предсказанных значений и реальных представлена в листинге 18.

Листинг 18 – функция для построения графиков

```
def plot_predictions(X, Y_original, Y_pred):
    fig, axes = plt.subplots(nrows=Y.shape[1], ncols=1, figsize=(10,
20))
    for i, ax in enumerate(axes):
        ax.scatter(X[:, 0], Y_original[:, i], color='blue',
label='Actual', s=10)

        ax.scatter(X[:, 0], Y_pred[:, i], color='red',
label='Predicted', s=10)

        ax.set_title(data.columns[i + 1])
        ax.legend()
    plt.tight_layout()
    plt.show()
```

В первую очередь, данная функция создает подграфики:

1) `plt.subplots` создает несколько подграфиков (ось `fig` для общей фигуры и `axes` для отдельных подграфиков);

2) `nrows=Y.shape[1]` задает количество строк подграфиков равным количеству выходных признаков (столбцов) в `Y`;

3) `ncols=1` задает количество столбцов подграфиков равным 1;

4) `figsize=(10, 20)` задает размер фигуры (ширина 10 и высота 20).

Отрисовка данных на подграфике:

5) Цикл `for i, ax in enumerate(axes)` проходит по каждому подграфику и индексу `i`.

6) `ax.scatter(X[:, 0], Y_original[:, i], color='blue', label='Actual', s=10)` - Отрисовывает реальные значения (синие точки) на подграфике.

7) `ax.scatter(X[:, 0], Y_pred[:, i], color='red', label='Predicted', s=10)` - Отрисовывает предсказанные значения (красные точки) на подграфике.

Вывод по третьей главе

В третьей главе были приведены средства реализации разрабатываемой системы, сформированы наборы данных для трех параметров оптимизации ВДПТ и описана предобработка данных, реализации моделей многослойного перцептрона и рекуррентной нейронной сети, а также функция для предсказания коэффициентов и функция построения графиков.

4. ЭКСПЕРИМЕНТАЛЬНАЯ ЧАСТЬ

4.1. Сравнение предсказанных значений на графиках

Загрузим набор данных для первого параметра и сравним предсказанные значения многослойного перцептрона и рекуррентной нейронной сети с реальными.

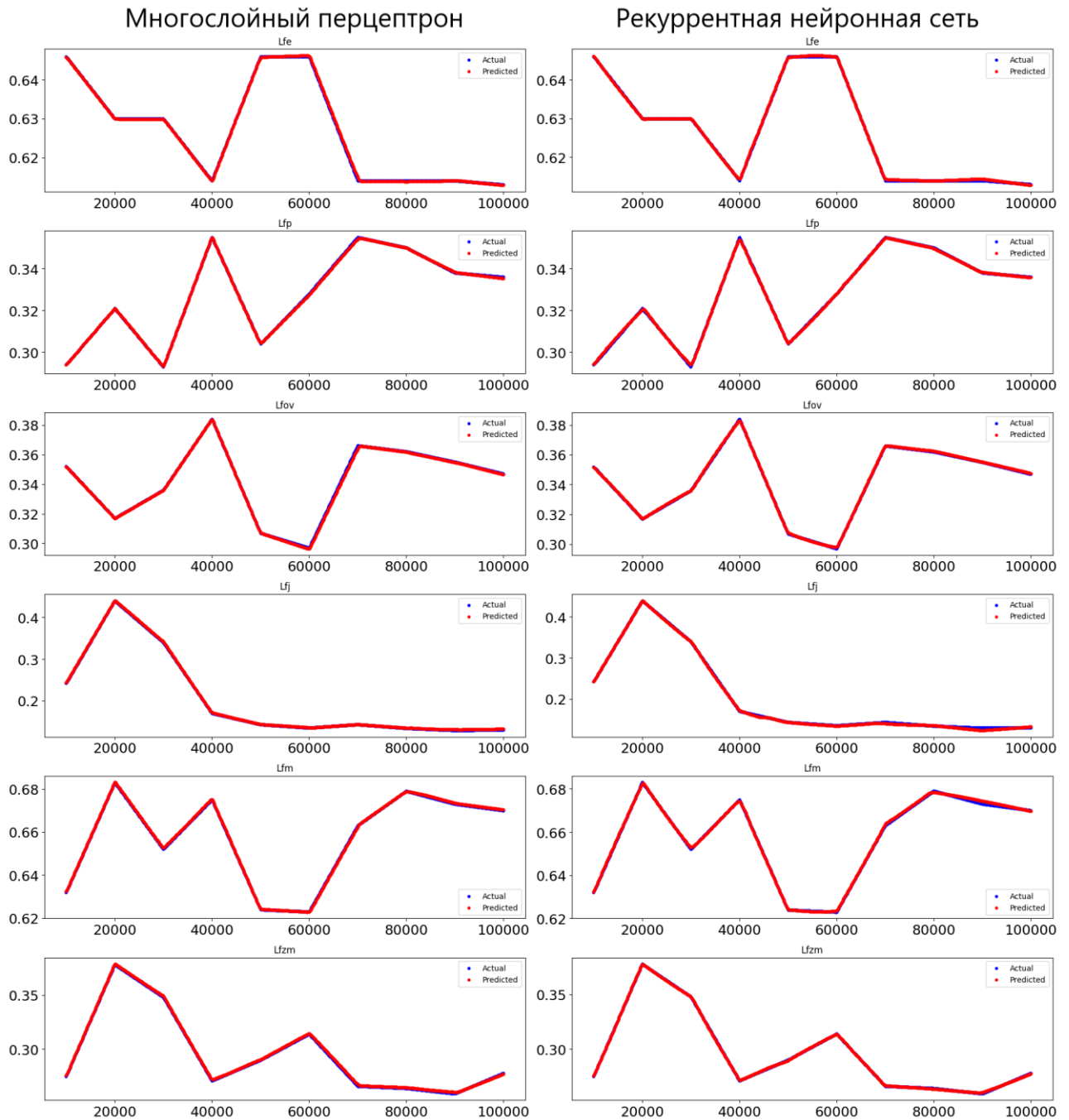


Рисунок 21 – Графики для минимальной массы активных материалов

На графиках изображены предсказанные значения для каждого параметра мощности в наборе данных, где синим отображаются реальные значения, а красным предсказанные. На рисунке 21 изображены предсказания для минимальной массы активных материалов, можно увидеть, что многослойный перцептрон справляется с предсказанием лучше.



Рисунок 22 - Графики для минимальной массы магнитов

На рисунке 22 изображены предсказания для минимальной массы магнитов. Обе модели справляются с предсказанием примерно одинаково.

На рисунке 23 изображены предсказания для минимального объема. На них можно увидеть, что рекуррентная нейронная сеть справляется с предсказанием лучше, чем многослойный перцептрон.

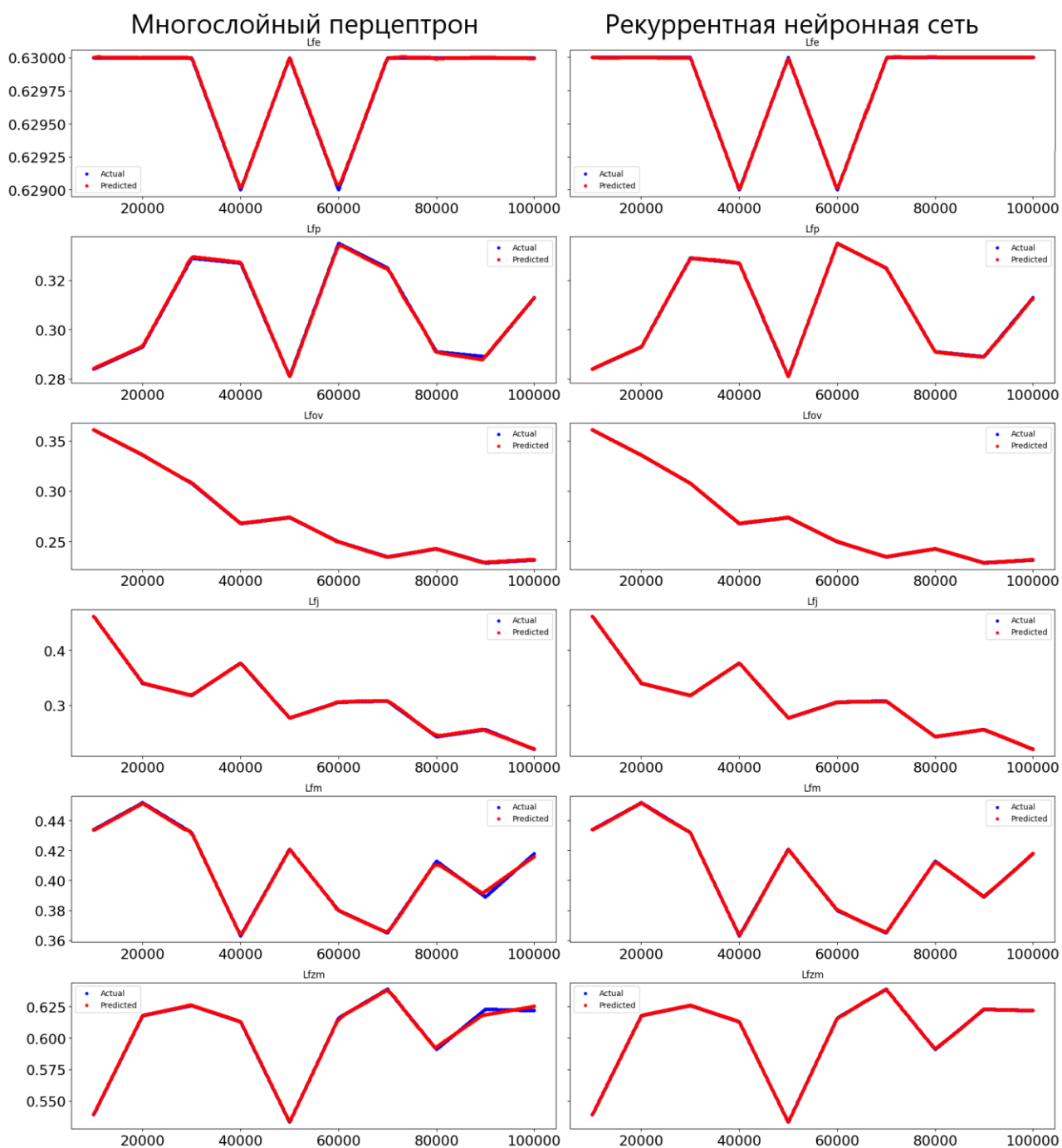


Рисунок 23 – Графики для минимального объема

Исходя из этих графиков можно увидеть, что в целом обе модели нейронной сети справляются с предсказанием значений, однако у многослойного перцептрона выше максимальное отклонение, чем у рекуррентной нейронной сети.

4.2. Тестирование функции предсказания

Пример работы функции предсказания изображен на рисунках 24-25 для многослойного перцептрона и рекуррентной нейронной сети соответственно. Функция предсказывает коэффициенты минимального объема для мощности двигателя 50000 Вт.

```
# Функция для предсказания на основе введенного значения X
def predict_y(input_x):
    input_x = scaler_x.transform(np.array(input_x).reshape(-1, 1)) # Нормализуем входные данные
    prediction = model.predict(input_x)
    prediction = scaler_y.inverse_transform(prediction) # Де-нормализуем выходные данные
    return prediction

# Пример использования функции
input_x = float(input('Введите мощность двигателя: '))
predicted_y = predict_y(input_x)
# Список названий для каждого элемента в predicted_y
output_names = ['Lfe', 'Lfp', 'Lfov', 'Lfj', 'Lfm', 'Lfzm']

# Вывод значений с их названиями
print(f'Input: {input_x}')
print('Predicted Output:')
for name, value in zip(output_names, predicted_y[0]):
    print(f'{name}: {value}')

Введите мощность двигателя: 50000
1/1 [=====] - 0s 29ms/step
Input: 50000.0
Predicted Output:
Lfe: 0.6299960613250732
Lfp: 0.2814805507659912
Lfov: 0.27368950843811035
Lfj: 0.27730777859687805
Lfm: 0.4206743836402893
Lfzm: 0.5340438485145569
```

Рисунок 24 – Предсказания коэффициентов многослойного перцептрона

```

# Функция для предсказания на основе введенного значения X
def predict_y(input_x):
    input_x = scaler_x.transform(np.array(input_x).reshape(-1, 1)) # Нормализуем входные данные
    prediction = model.predict(input_x)
    prediction = scaler_y.inverse_transform(prediction) # Де-нормализуем выходные данные
    return prediction

# Пример использования функции
input_x = float(input('Введите мощность двигателя: '))
predicted_y = predict_y(input_x)
# Список названий для каждого элемента в predicted_y
output_names = ['Lfe', 'Lfp', 'Lfov', 'Lfj', 'Lfm', 'Lfzm']

# Вывод значений с их названиями
print(f'Input: {input_x}')
print('Predicted Output:')
for name, value in zip(output_names, predicted_y[0]):
    print(f'{name}: {value}')

```

Введите мощность двигателя: 50000
1/1 [=====] - 0s 41ms/step
Input: 50000.0
Predicted Output:
Lfe: 0.6299875378608704
Lfp: 0.281186580657959
Lfov: 0.2742675244808197
Lfj: 0.27734678983688354
Lfm: 0.4203900694847107
Lfzm: 0.5335437059402466

Рисунок 25 – Предсказание коэффициентов рекуррентной нейронной сети

4.3. Инженерная методика проектирования ВЭЖД на основе коэффициентов

Программа расчета двигателя комбинированного возбуждения.

1. Внутренний диаметр, сердечника ротора (м):

$D_v = D \cdot \sqrt{1 - fs}$, где D – диаметр якоря, (м); fs – независимая переменная.

$$D_v = 0,21 \cdot \sqrt{1 - 1} = 0$$

2. Высота паза, (м):

$h_{paza_} = \frac{D}{2 \cdot 2} \cdot (1 - \sqrt{1 - fe \cdot fs})$, где fe – независимая переменная.

$$h_{paza_} = \frac{0,21}{2 \cdot 2} \cdot (1 - \sqrt{1 - 0,63 \cdot 1}) = 0,021$$

3. Ширина паза, (м):

$b_{paza} = \frac{fp \cdot fe \cdot fs \cdot \pi \cdot D}{2 \cdot (1 - \sqrt{1 - fe \cdot fs}) \cdot z}$, где fp – независимая переменная; z – число пазов на

статоре; π – число Пи.

$$b_{paza} = \frac{0,293 \cdot 0,63 \cdot 1 \cdot 3,14 \cdot 0,21}{2 \cdot (1 - \sqrt{1 - 0,63 \cdot 1}) \cdot 48} = 3,238 \cdot 10^{-3}$$

4. Высота окна обмотки возбуждения, (м):

$$h_{ov} = \frac{D}{2} \left[\sqrt{fe \cdot fs + fov \cdot (2 \cdot \sqrt{1 - fe \cdot fs} - 1) - \sqrt{fe \cdot fs}} \right], \quad \text{где } fov -$$

независимая переменная.

$$h_{ov} = \frac{0,21}{2} \left[\sqrt{0,63 \cdot 1 + 0,336 \cdot (2 \cdot \sqrt{1 - 0,63 \cdot 1} - 1) - \sqrt{0,63 \cdot 1}} \right] = 0,01707$$

5. Сумма высоты магнита и индуктора, (м):

$$h_j h_m = \frac{D}{2} (\sqrt{1 + fj \cdot fs} - 1), \quad \text{где } fj - \text{ независимая переменная.}$$

$$h_j h_m = \frac{0,21}{2} (\sqrt{1 + 0,34 \cdot 1} - 1) = 0,017$$

6. Наружный диаметр, (м):

$$D_n = D \cdot \sqrt{1 + fj \cdot fs},$$

$$D_n = 0,21 \cdot \sqrt{1 + 0,34 \cdot 1} = 0,24309$$

7. Высота магнита, (м):

$$h_m = \frac{D}{2} (\sqrt{1 + fj \cdot fs \cdot fm} - 1), \quad \text{где } fm - \text{ независимая переменная.}$$

$$h_m = \frac{0,21}{2} (\sqrt{1 + 0,34 \cdot 1 \cdot 0,652} - 1) = 0,01105612$$

8. Высота спинки индуктора, (м):

$$h_j = \frac{D}{2} (\sqrt{1 + fj \cdot fs} - \sqrt{1 + fj \cdot fs \cdot fm}),$$

$$h_j = \frac{0,21}{2} (\sqrt{1 + 0,34 \cdot 1} - \sqrt{1 + 0,34 \cdot 1 \cdot 0,652}) = 5,49 \cdot 10^{-3}$$

Где h_{ov} – высота обмотки возбуждения, м; $h_{паза}$ – высота паза, м.

$$\frac{D}{2} - \frac{D_v}{2} = h_{ov} + h_{паза},$$

$$\frac{0,21}{2} - \frac{0}{2} = 0,017 + 0,021 = 0,056$$

9. Высота спинки якоря, (м):

$$h_a = \frac{D}{2} \left[1 - \sqrt{1 - fs} - (1 - \sqrt{1 - fe \cdot fs}) - \right.$$

$$\left. \left[\sqrt{fe \cdot fs + fov \cdot (2\sqrt{1 - fe \cdot fs} + fe \cdot fs - 1) - \sqrt{fe \cdot fs}} \right] \right],$$

$$h_a = \frac{0,21}{2} \left[1 - \sqrt{1 - 1} - (1 - \sqrt{1 - 0,63 \cdot 1}) - \right. \\ \left. \left[\sqrt{0,63 \cdot 1 + 0,336 \cdot (2\sqrt{1 - 0,63 \cdot 1} + 0,63 \cdot 1 - 1) - \sqrt{0,63 \cdot 1}} \right] \right] = 0,046802$$

10. Ширина магнита, (м):

$$b_{m_1} = \frac{\pi \cdot D \cdot fzm \cdot (1 + fj \cdot fs \cdot fm)}{N_{pol} \cdot 2}, \quad \text{где } fzm - \text{ коэффициент корреляции;}$$

N_{pol} – исходные данные.

$$b_{m_1} = \frac{3,14 \cdot 0,21 \cdot 0,348 \cdot (1 + 0,34 \cdot 1 \cdot 0,652)}{6 \cdot 2} = 0,0151$$

11. Площадь голой меди в пазе (м²):

$S_{cu} = h_{paза} \cdot b_{paза} \cdot z \cdot f_{cu}$, где f_{cu} – коэффициент меди; $b_{paза}$ – ширина паза, м.

$$S_{cu} = 0,019 \cdot 0,07 \cdot 48 \cdot 0,3 = 1,915 \cdot 10^{-3}$$

$$S_{cu_} = \frac{fp \cdot fe \cdot fs \cdot \pi \cdot D \cdot f_{cu}}{2 \cdot (\sqrt{1 - fe \cdot fs})} \cdot \frac{D}{2} [(1 - \sqrt{1 - fe \cdot fs})], \quad (4.13)$$

$$S_{cu_} = \frac{0,293 \cdot 0,63 \cdot 1 \cdot 3,14 \cdot 0,21 \cdot 0,3}{2 \cdot (\sqrt{1 - 0,63 \cdot 1})} \cdot \frac{0,21}{2} [(1 - \sqrt{1 - 0,63 \cdot 1})] = 1,918 \cdot 10^{-3}$$

12. Число зубцов на статоре:

$$z_1 = mf \cdot q \cdot N_{pol},$$

где q – число пазов на полюс и фазу.

$$z_1 = 3 \cdot 1 \cdot 6 = 18$$

13. Плотность тока, (А/м²):

$$j_a = 4,48 \cdot 10^6 \cdot 2,3 = 1,03 \cdot 10^7$$

14. Линейная нагрузка, (А/м):

$$A_{lin} = \frac{S_{cu_} \cdot j_a \cdot \frac{(mf-1)}{mf}}{(\pi \cdot D)}, \quad \text{где } S_{cu_} - \text{ площадь голой меди в пазе, м}^2.$$

$$A_{lin} = \frac{1,918 \cdot 10^{-3} \cdot 1,03 \cdot 10^7 \cdot \frac{(3-1)}{3}}{(3,14 \cdot 0,21)} = 1,997 \cdot 10^4$$

15. Активная длина (м):

$$L_{del} = \frac{P_n \frac{60 \cdot ke}{\pi^2 \cdot k_{выпр}} \cdot (mf-1)}{\alpha \cdot (kb \cdot kob) \cdot B_{delta} \cdot A_{lin} \cdot D^2 \cdot n},$$

где ke – коэффициент ЭДС якоря; kob – обмоточный коэффициент обмотки статора; kb – коэффициент формы поля; A_{lin} – линейная нагрузка, А/м; $k_{выпр}$ – коэффициент выпрямления.

$$L_{del} = \frac{8500 \frac{60 \cdot 1,03}{3,14^2 \cdot 1,654} \cdot (3-1)}{0,8 \cdot (1,11 \cdot 0,95) \cdot 0,586 \cdot 1,997 \cdot 10^4 \cdot 0,21^2 \cdot 3000} = 0,173282$$

16. ЭДС фазы (В):

$$EDSf = \frac{U_n \cdot ke}{k_{выпр}},$$

где U_n – фазное напряжение В.

$$EDSf = \frac{28 \cdot 1,02632}{1,654} = 17,374$$

17. Номинальный ток (А):

$$I_{nom} = \frac{P_n}{EDSf \cdot \left(\frac{k_{выпр}}{ke}\right)},$$

где P_n – номинальная выходная мощность (Вт).

$$I_{nom} = \frac{8500}{17,374 \cdot \left(\frac{1,654}{1,02632}\right)} = 1,071 \cdot 10^3$$

18. Число проводников у провода:

$$un = \frac{A_{lin} \cdot (\pi \cdot D)}{I_{nom} \cdot z1 \cdot \frac{mf-1}{mf}},$$

$$un = \frac{1,997 \cdot 10^4 \cdot (3,14 \cdot 0,21)}{1,071 \cdot 10^3 \cdot 18 \cdot \frac{3-1}{3}} = 1,025 \cdot 10^3$$

19. Число витков в фазе:

$$wf = \frac{un \cdot z1}{2 \cdot mf},$$

$$wf = \frac{1,025 \cdot 10^3 \cdot 18}{2 \cdot 3} = 3,074$$

20. ЭДС фазы (В):

$$E_f = 4 \cdot kb \cdot kob \cdot f \cdot wf \left(\alpha \cdot B_{delta} \frac{\pi \cdot D}{N_{pol}} \cdot L_{del} \right), \text{ где } L_{del} \text{ – активная длина,}$$

м; B_{delta} – значение индукции (Тл) f – частота питающего напряжения, В.

$$E_f = 4 \cdot 1,11 \cdot 0,95 \cdot 150 \cdot 3,074 \left(0,8 \cdot 0,586 \frac{3,14 \cdot 0,21}{6} \cdot 0,173 \right) = 17,374$$

21. Расчетная мощность (Вт):

$$P_{rash} = \frac{\pi^2}{60} \cdot \frac{k_{выпр}}{(mf-1) \cdot ke} \cdot kb \cdot kob \cdot (alfa \cdot B_{delta}) \cdot A_{lin} \cdot D^2 \cdot L_{del} \cdot n,$$

$$P_{rash} = \frac{3,14^2}{60} \cdot \frac{1,654}{(3-1) \cdot 1,02632} \cdot 1,11 \cdot 0,95 \cdot (0,8 \cdot 0,586) \cdot 1,997 \cdot 10^4$$

$$\times 0,21^2 \cdot 0,173282 \cdot 3000 = 3 \cdot 10^4$$

22. Выпрямленное напряжение(В):

$$U_d = \frac{E_f \cdot k_{выпр}}{ke},$$

где E_f – ЭДС фазы.

$$U_d = \frac{17,374 \cdot 1,654}{1,02632} = 28$$

23. Выпрямление тока(А):

$$I_d = \frac{P_{rash}}{U_d}, \text{ где } P_{rash} - \text{ расчетная мощность, Вт; } U_d - \text{ выпрямленное}$$

напряжение, В.

$$I_d = \frac{3 \cdot 10^4}{28} = 1,071 \cdot 10^3$$

24. Постоянная двигателя:

$$C_E = \frac{N_{pol} \cdot \left(\frac{z \cdot un \cdot 2}{3}\right)}{\pi \cdot a},$$

$$C_E = \frac{6 \cdot \left(\frac{48 \cdot 1,025 \cdot 10^3 \cdot 2}{3}\right)}{3,14 \cdot 1} = 31,315$$

25. Поток полюса:

$$Potok = alfa \cdot B_{delta} \cdot \frac{\pi \cdot D}{N_{pol}} \cdot L_{del},$$

$$Potok = 0,8 \cdot 0,586 \cdot \frac{3,14 \cdot 0,21}{6} \cdot 0,173 = 8,932 \cdot 10^{-3}$$

26. Номинальный момент (Нм):

$$M_- = \frac{I_d \cdot U_d \cdot ke}{\left(\frac{2 \cdot \pi \cdot n}{60}\right)},$$

$$M_- = \frac{1,071 \cdot 10^3 \cdot 28 \cdot 1,02632}{\left(\frac{2 \cdot 3,14 \cdot 3000}{60}\right)} = 98,006$$

$$M_n = C_E \cdot Potok \cdot Inom,$$

$$M_n = 31,315 \cdot 8,932 \cdot 10^{-3} \cdot 1,071 \cdot 10^3 = 299,689.$$

Где I_{nom} – номинальный ток, А; C_E – постоянная двигателя; $Potok$ – поток полюса.

4.3 Расчет параметра обмотки статора

1. Длина лобовой части обмотки якоря(м):

$$L_{\text{лоб}} = 1,7 \cdot (D - 2 \cdot h_{\text{паза}}) \cdot \left(\frac{\pi}{N_{\text{пол}}} \right), \text{ где } h_{\text{паза}} - \text{высота паза, м.}$$

$$L_{\text{лоб}} = 1,7 \cdot (0,21 - 2 \cdot 0,021) \cdot \left(\frac{3,14}{6} \right) = 0,15$$

2. Средняя длина витки обмотки якоря(м):

$$L_{\text{ср}} = 2 \cdot (L_{\text{лоб}} + L_{\text{дел}}), \text{ где } L_{\text{лоб}} - \text{длина лобовой части обмотки якоря, м; } L_{\text{дел}}$$

– активная длина, м.

$$L_{\text{ср}} = 2 \cdot (0,15 + 0,173) = 0,647$$

3. Сечение эффективного проводника обмотки якоря (м^2):

$$q_{\text{эф}} = \frac{S_{\text{св}}}{z_1 \cdot u_n}, \text{ где } u_n - \text{число проводников у провода; } z_1 - \text{число зубцов}$$

ротора.

$$q_{\text{эф}} = \frac{1,918 \cdot 10^{-3}}{18 \cdot 1,025 \cdot 10^3} = 1,04 \cdot 10^{-4}$$

4. Удельное сопротивление при 75 градусов:

$$p_{75} = 2,184 \cdot 10^{-8}$$

5. Сопротивление обмотки якоря (Ом):

$$r_{a75} = p_{75} \cdot L_{\text{ср}} \cdot \frac{w_f}{q_{\text{эф}} \cdot a}, \text{ где } q_{\text{эф}} - \text{сечение эффективного проводника обмотки}$$

якоря; a – число параллельных ветвей.

$$r_{a75} = 2,184 \cdot 10^{-8} \cdot 0,647 \cdot \frac{3,074}{1,04 \cdot 10^{-4} \cdot 1} = 4,179 \cdot 10^{-4}$$

6. Напряжение в фазе(В):

$$U_f = E_f - r_{a75} \cdot I_{\text{ном}}, \text{ где } I_{\text{ном}} - \text{номинальный ток, А; } E_f - \text{ЭДС фазы.}$$

$$U_f = 17,374 - 4,179 \cdot 10^{-4} \cdot 1,071 \cdot 10^3 = 16,926$$

7. Коэффициент ЭДС:

$$k_{e-} = \frac{E_f}{U_f},$$

$$k_{e-} = \frac{17,374}{16,926} = 1,02645$$

8. Электрические потери якоря (Вт):

$$P_{eala} = I_{nom}^2 \cdot r_{a75} \cdot (mf - 1),$$

$$P_{eala} = 1,071 \cdot 10^{32} \cdot 4,179 \cdot 10^{-4} \cdot (3 - 1) = 959,47$$

9. Константы:

Где $a_{\delta} = 0,8$ – коэффициент полюсного перекрытия; $\delta_{\text{зазор}} = 1 \cdot 10^3$ – воздушный зазор (м); $b_{\text{паз}} = 7 \cdot 10^{-3}$ – ширина паза(м); $\mu_0 = 1,257 \cdot 10^{-6} \cdot \frac{m \cdot kg}{s^2 \cdot A^2}$ – удельная магнитная проводимость; $k_c = 0,96$ – коэффициент заполнения пакета сталью.

$$\tau = \frac{\pi \cdot D}{N_{pol}},$$

$$\tau = \frac{3,14 \cdot 0,21}{6} = 0,11, \text{ где } \tau \text{ – допустимое напряжение материала при кручении}$$

(Па).

$$t_{z1} = \frac{\pi \cdot D}{z_1} = 0,037$$

$$t_{z1} = \frac{3,14 \cdot 0,21}{18} = 0,037$$

4.4. Расчет магнитной цепи

1. Магнитный поток, (Вт):

$$\Phi_{\text{маг}} = a_{\delta} \cdot B_{\text{норм}} \cdot \tau \cdot \frac{L_{\delta}}{2}, \text{ где } B_{\text{норм}} \text{ – магнитный поток от постоянного}$$

магнита; L_{δ} – расчётная длина магнитопровода, м.

$$\Phi_{\text{маг}} = 0,8 \cdot 0,586 \cdot 0,11 \cdot \frac{0,173}{2} = 4,466 \cdot 10^{-3}$$

$$\Phi = a_{\delta} \cdot B_{\text{норм}} \cdot \tau \cdot L_{\delta},$$

$$\Phi = 0,8 \cdot 0,586 \cdot 0,11 \cdot 0,173 = 8,932 \cdot 10^{-3}$$

2. МДС воздушного зазора, (А):

$$k_{\delta} = \frac{t_{z1} + 10 \cdot \delta_{\text{зазор}}}{0,037 - b_{\text{паз}} + 10 \cdot \delta_{\text{зазор}}},$$

$$k_{\delta} = \frac{0,037 + 10 \cdot 1 \cdot 10^3}{0,037 - 7 \cdot 10^{-3} + 10 \cdot 1 \cdot 10^3} = 1,177, \text{ где } \delta_{\text{зазор}} \text{ – воздушный зазор между}$$

статором и ротором, м; μ_0 – удельная магнитная проводимость вакуума, Гн/м.

$$F_{\delta} = \frac{1}{\mu_0} \cdot B_{\delta\text{ном}} \cdot \delta_{\text{зазор}} \cdot k_{\delta},$$

$$F_{\delta} = \frac{1}{1,257 \cdot 10^{-6}} \cdot 0,586 \cdot 1 \cdot 10^3 \cdot 1,177 = 548,925$$

3. МДС зубцовой зоны (сталь 2212), (А):

$$t_{Z\text{серединапаз}} = \frac{\pi(D_{\text{статора}} - h_{\text{пазполн}})}{Z},$$

$$t_{Z\text{серединапаз}} = \frac{3,14(0,21 - 0,021)}{18} = 0,033, \text{ где } h_{\text{пазполн}} - \text{полная высота паза, м.}$$

$$k_{Z\text{зоны}} = \frac{b_{\text{паз}} \cdot L_{\delta}}{(t_{Z\text{серединапаз}} - b_{\text{паз}}) \cdot L_{\delta} \cdot k_c}, \text{ где } k_c - \text{коэффициент спинки якоря.}$$

$$k_{Z\text{зоны}} = \frac{7 \cdot 10^{-3} \cdot 0,173}{(0,033 - 7 \cdot 10^{-3}) \cdot 0,173 \cdot 0,96} = 0,28$$

4. Индукция в спинке якоря, (Тл):

$$B_{Z\text{середина}} = \frac{B_{\delta\text{ном}} \cdot t_{z1} \cdot L_{\delta}}{(t_{Z\text{серединапаз}} - b_{\text{паз}}) \cdot L_{\delta} \cdot k_c}, \text{ где } t_{Z\text{серединапаз}} - \text{МДС зубцовой зоны, А.}$$

$$B_{Z\text{середина}} = \frac{0,586 \cdot 0,173 \cdot 0,037}{(0,033 - 7 \cdot 10^{-3}) \cdot 0,173 \cdot 0,96} = 0,858$$

$$H_Z = \text{Нст2212} \left(\frac{B_{Z\text{середина}}}{k_{Z\text{зоны}}} \right), \text{ где } B_{Z\text{середина}} - \text{индукция в спинке якоря, Тл.}$$

$$H_Z = \text{Нст2212} \left(\frac{0,858}{0,28} \right) = 3,686 \cdot 10^5$$

$$F_Z = 2 \cdot h_Z \cdot H_Z,$$

$$F_Z = 2 \cdot 0,021 \cdot 3,686 \cdot 10^5 = 1,516 \cdot 10^4$$

5. МДС спинке статора (сталь 2112), (А):

$$t_{Z\text{днопаз}} = \frac{\pi(D_{\text{статора}} - 2 \cdot h_{\text{пазполн}})}{Z}, \text{ где } h_{\text{пазполн}} - \text{полная высота паза, м; } Z - \text{число}$$

пазов ротора.

$$t_{Z\text{днопаз}} = \frac{3,14(0,21 - 2 \cdot 0,021)}{18} = 0,029$$

$$B_a = \frac{B_{\delta\text{ном}} \cdot t_{z1} \cdot L_{\delta}}{t_{Z\text{днопаз}} \cdot L_{\delta} \cdot k_c}, \text{ где } k_c - \text{коэффициент заполнения пакета сталью.}$$

$$B_a = \frac{0,586 \cdot 0,173 \cdot 0,037}{0,029 \cdot 0,173 \cdot 0,96} = 0,759$$

$$H_a = \text{Нст2212} \cdot (B_a),$$

$$H_a = \text{Нст2212} \cdot 0,759 = 122,001$$

$$F_a = 2 \cdot h_a \cdot H_a, \text{ где } h_a - \text{высота спинки якоря, м}$$

$$F_a = 2 \cdot 0,005 \cdot 122,001 = 1,22$$

6. МДС стенки статора у ОВ пару полюсов (сталь10), (А):

$$h_{ОВ} = 0,017$$

$$b_{ОВ} = 0,025$$

Где $b_{ОВ}$ – высота окна обмотки возбуждения, м.

$$b_{стенки} = \frac{\left[D_{статора} - 2(2 \cdot h_{пазполн} + h_a) - \frac{h_{ОВ}^2}{2} \right]}{Z}, \text{ где } h_a \text{ – высота спинки якоря, м; } h_{ОВ} \text{ –}$$

высота катушки обмотки возбуждения, м.

$$b_{стенки} = \frac{\left[0,21 - 2(2 \cdot 0,021 + 0,005) - \frac{0,017^2}{2} \right]}{18} = 0,025$$

7. Индукция в стенке якоря, (Тл):

$$B_{стенки} = \frac{B_{\delta ном} \cdot t_{z1} \cdot L_{\delta}}{b_{стенки} \cdot \left(\frac{L_{del} + 0,06 - b_{ОВ}}{k_{Z зоны}} \right)}, \text{ где } b_{ОВ} \text{ – ширина катушки обмотки}$$

возбуждения, м; L_{del} – активная длина, м. $k_{Z зоны}$ – коэффициент МДС зубцовой зоны, А.

$$B_{стенки} = \frac{0,586 \cdot 0,037 \cdot 0,173}{0,025 \cdot \left(\frac{0,173 + 0,06 - 0,025}{0,28} \right)} = 1,44$$

$$H_{стенки} = \text{Нст}10 \left(\frac{B_{стенки}}{1} \right), \text{ где } B_{стенки} \text{ – индукция в стенке якоря.}$$

$$H_{стенки} = \text{Нст}10 \left(\frac{1,44}{1} \right) = 1,996 \cdot 10^3$$

$$F_{стенки} = 2 \cdot h_{ОВ} \cdot H_{стенки},$$

$$F_{стенки} = 2 \cdot 0,017 \cdot 1,996 \cdot 10^3 = 68,127$$

8. МДС трубы статора на пару полюсов (сталь10), (А):

$$r_{наруж_{тр.ст}} = \frac{D_{статора} - 2(h_{пазполн} + h_a + h_{ОВ})}{2}, \text{ где } h_a \text{ – высота спинки якоря, м.}$$

$$r_{наруж_{тр.ст}} = \frac{0,21 - 2(0,021 + 0,005 + 0,017)}{2} = 0,062$$

$$r_{внутр_{тр.ст}} = \frac{D_v}{2},$$

$$r_{внутр_{тр.ст}} = \frac{0}{2} = 0$$

$$S_{тр.ст} = \pi \cdot \left(r_{наруж_{тр.ст}}^2 - r_{внутр_{тр.ст}}^2 \right),$$

$$S_{тр.ст} = 3,14 \cdot (0,062^2 - 0^2) = 0,012$$

$B_{\text{тр.ст}} = \frac{\Phi \cdot p}{S_{\text{тр.ст}}}$, где p – число полюсов; L_{del} – активная длина, м.

$$B_{\text{тр.ст}} = \frac{8,932 \cdot 10^{-3} \cdot 3}{0,012} = 2,193$$

$$H_{\text{стст}} = \text{Нст}10(B_{\text{тр.ст}}),$$

$$H_{\text{стст}} = \text{Нст}10(B_{\text{тр.ст}}) = 5,821 \cdot 10^4$$

$$F_{\text{тр.ст}} = L_{del} \cdot H_{\text{стст}} \cdot 1,$$

$$F_{\text{тр.ст}} = 0,173 \cdot H_{\text{стст}} \cdot 1 = 1,009 \cdot 10^4$$

9. МДС сердечника полюса (магнита) на пару полюсов (сталь10), (А):

$$B_m = 0,586$$

$$H_m = 284,4$$

$F_m = 2h_{\text{маг}} \cdot H_m$, где $h_{\text{маг}}$ – высота магнита, м.

$$F_m = 2 \cdot 0,011 \cdot 284,4 = 6,289$$

10. Расчет геометрии, (м):

$r_{\text{б.рот}} = \frac{D_n}{2}$, где D_n – наружный диаметр, м.

$$r_{\text{б.рот}} = \frac{0,243}{2} = 0,122$$

$r_{\text{м.рот}} = \frac{D_n - 2 \cdot h_{\text{спинкирот}}}{2}$, где $h_{\text{спинкирот}}$ – высота спинки ротора, м.

$$r_{\text{м.рот}} = \frac{0,243 - 2 \cdot 0,02}{2} = 0,102$$

$$S_{\text{тр.рот}} = (r_{\text{б.рот}}^2 - r_{\text{м.рот}}^2),$$

$$S_{\text{тр.рот}} = (0,122^2 - 0,102^2) = 0,014$$

$$B_{\text{тр.рот}} = \frac{\Phi \cdot p}{S_{\text{тр.рот}}},$$

$$B_{\text{тр.рот}} = \frac{8,932 \cdot 10^{-3} \cdot 3}{0,014} = 1,912$$

$$H_{\text{тр.рот}} = \text{Нст}10(B_{\text{тр.рот}}),$$

$$H_{\text{тр.рот}} = \text{Нст}10(B_{\text{тр.рот}}) = 1,536 \cdot 10^4$$

11. МДС ярма ротора (сталь 10), (А):

$$F_{\text{тр.рот}} = L_{del} \cdot H_{\text{тр.рот}},$$

$$F_{\text{тр.рот}} = 0,173 \cdot 1,536 \cdot 10^4 = 2,662 \cdot 10^3$$

$$F_Z = 1516 \cdot 1 \cdot F_{\text{тр.ст}},$$

$$\sum F = F_\delta + F_Z + F_a + F_{\text{стенки}} + F_{\text{тр.ст}} + F_m + F_{\text{тр.рот}},$$

$$\sum F = 548,925 + 1,009 \cdot 10^4 + 1,22 + 68,127 + 1,009 \cdot 10^4 + 6,289 + 2,662 \cdot 10^3 = 2,854 \cdot 10^4$$

$$k_\mu = \frac{\sum F}{F_\delta},$$

$$k_\mu = \frac{2,854 \cdot 10^4}{548,925} = 51,98$$

4.5. Расчет обмотки возбуждения

1. Диаметр средний обмотки возбуждения, (м):

$D_{\text{ср.ОВ}} = D_{\text{статора}} - 2(h_Z + h_a) - h_{\text{ОВ}}$, где $D_{\text{статора}}$ – диаметр статора, м; $h_{\text{ОВ}}$ – высота катушки обмотки возбуждения, м.

$$D_{\text{ср.ОВ}} = 0,21 - 2(0,021 + 0,005) - 0,017 = 0,142$$

2. Удельное сопротивление меди при 20 градусах:

$$\rho_{20} = 1,79 \cdot 10^{-8}$$

3. Средняя длина витка, (м):

$L_{\text{ср.ОВ}} = \pi \cdot D_{\text{ср.ОВ}}$, где $D_{\text{ср.ОВ}}$ – диаметр средний обмотки возбуждения, м.

$$L_{\text{ср.ОВ}} = 3,14 \cdot 0,142 = 0,445$$

4. Рабочая температура, (°C):

$$t_{\text{ОВ}} = 75$$

5. Удельное сопротивление меди при 75 градусах:

$\rho_{75} = \rho_{20}[1 + 0,004(t_{\text{ОВ}} - 20)]$, где $t_{\text{ОВ}}$ – рабочая температура.

$$\rho_{75} = 1,79 \cdot 10^{-8}[1 + 0,004(75 - 20)] = 2,184 \cdot 10^{-8}$$

6. Напряжение на обмотке возбуждения, (В):

$U_{\text{ов}} = \frac{U_n}{2}$, где U_n – номинальное фазное напряжение В.

$$U_{\text{ов}} = \frac{28}{2} = 14$$

7. Сила намагничивания обмотки возбуждения (А):

$$F_{\text{ов}} = 2,854 \cdot 10^4$$

8. Площадь поперечного сечения проводника обмотки возбуждения, (м²):

$$q_{\text{ов}} = \frac{F_{\text{ов}} \cdot \rho_{75} \cdot L_{\text{ср.ОВ}}}{U_{\text{ов}}},$$

$$q_{ov} = \frac{2,854 \cdot 10^4 \cdot 2,184 \cdot 10^{-8} \cdot 0,445}{14} = 1,983 \cdot 10^{-5}, \text{ где } L_{cpOB} \text{ — средняя длина}$$

обмотки возбуждения, м

9. Диаметр провода обмотки возбуждения, (м):

$$d_{OB} = \sqrt{\frac{4 \cdot q_{ov}}{\pi}},$$

$$d_{OB} = \sqrt{\frac{4 \cdot 1,983 \cdot 10^{-5}}{3,14}} = 5,025 \cdot 10^{-3}$$

10. Плотность тока в обмотке возбуждении (A/m^2):

$$i_{ov} = 5 \cdot 10^6$$

$$k_{cuov} = 0,5$$

11. Ток в обмотке возбуждения(А):

$$I_{ov} = i_{ov} \cdot q_{ov},$$

$$I_{ov} = 5 \cdot 10^6 \cdot 1,983 \cdot 10^{-5} = 99,143$$

12. Число векторов:

$$W_{ov} = \frac{F_{ov}}{I_{ov}},$$

$$W_{ov} = \frac{2,854 \cdot 10^4}{99,143} = 287,83$$

13. Расчет сопротивления обмотки возбуждения, (Ом):

$$r_{OB} = \frac{\rho_{75} \cdot L_{cpOB} \cdot W_{ov}}{q_{ov}} = 0,141$$

$$r_{OB} = \frac{2,184 \cdot 10^{-8} \cdot 0,445 \cdot 287,83}{1,983 \cdot 10^{-5}} = 0,141, \text{ где } q_{ov} \text{ — площадь поперечного сечения}$$

проводника обмотки возбуждения, m^2 ; L_{cpOB} — средняя длина обмотки возбуждения, м.

$$S_{cuov} = W_{ov} \cdot q_{ov},$$

$$S_{cuov} = 287,83 \cdot 1,983 \cdot 10^{-5} = 5,707 \cdot 10^{-3}$$

14. Ширина окна обмотки возбуждения, (м):

$$b_{ov} = 0,669$$

15. Высота обмотки возбуждения, (м):

$$h_{OB} = 0,017$$

16. Площадь полюса (m^2):

$$S_{\Pi} = \tau \cdot a_{\delta} \frac{L_{\delta}}{2},$$

$$S_{\Pi} = 0,11 \cdot 0,8 \frac{0,173}{2} = 7,621 \cdot 10^{-3}$$

17. Магнитное сопротивление зазора (Гн/м):

$$R_{\delta} = \frac{\delta_{\text{зазор}} \cdot k_{\delta}}{\mu_0 \cdot S_{\Pi}},$$

$$R_{\delta} = \frac{0,001 \cdot 1,177}{1,257 \cdot 10^{-6} \cdot 7,621 \cdot 10^{-3}} = 1,229 \cdot 10^5$$

18. Магнитное сопротивление зазора магнитной цепи (Гн/м):

$$\sum R = R_{\delta} \cdot k_{\mu} = 6,39 \cdot 10^6$$

$$\sum R = 1,229 \cdot 10^5 \cdot 51,98 = 6,39 \cdot 10^6$$

19. Поток обмотки возбуждения (Вб):

$$\Phi_{\text{ОВ}} = \frac{F_{\text{ОВ}}}{\sum R},$$

$$\Phi_{\text{ОВ}} = \frac{2,854 \cdot 10^4}{6,39 \cdot 10^6} = 4,466 \cdot 10^{-3}$$

20. Поток магнита (Вб):

$$\Phi_{\text{маг}} = 4,466 \cdot 10^{-3}$$

21. Суммарный поток (Вб):

$$\Phi = \Phi_{\text{маг}} + \Phi_{\text{ОВ}} = 8,932 \cdot 10^{-3}$$

$$\Phi = 4,466 \cdot 10^{-3} + 4,466 \cdot 10^{-3} = 8,932 \cdot 10^{-3}$$

4.6. Расчет массы активных материалов

1. Масса зубцов статора, (кг):

$$m_z = 7800 \cdot L_{\delta} \cdot k_c \cdot h_z (t_{z\text{серединапаз}} - b_{\text{паз}}) \cdot Z, \text{ где } k_c =$$

0,96 – коэффициент заполнения пакета сталью.

$$m_z = 7800 \cdot 0,173 \cdot 0,96 \cdot 0,021 (0,033 - 7 \cdot 10^{-3}) \cdot 18 = 12,51$$

2. Масса ярма статора, (кг):

$$m_a = 7800 \cdot L_{\delta} \cdot k_c \cdot \pi \left[\left(\frac{D_{\text{статор}} - 2h_z}{2} \right)^2 - \left[\frac{D_{\text{статор}} - 2(h_z + h_a)}{2} \right]^2 \right],$$

$$m_a = 7800 \cdot 0,173 \cdot 0,96 \cdot 3,14 \left[\left(\frac{0,21 - 2 \cdot 0,021}{2} \right)^2 - \left[\frac{0,21 - 2(0,021 + 0,005)}{2} \right]^2 \right] =$$

3,34

3. Масса меди обмотки статора, (кг):

$$m_{cu} = 8900 \cdot q_{эф} \cdot un \cdot z \cdot \frac{L_{ср}}{2},$$

$$m_{cu} = 8900 \cdot 1,04 \cdot 10^{-4} \cdot 1,025 \cdot 10^3 \cdot 18 \cdot \frac{0,647}{2} = 5,524$$

4. Масса меди обмотки возбуждения, (кг):

$$m_{ОВ} = 8900 \cdot q_{ОВ} \cdot L_{срОВ} \cdot w_{ОВ},$$

$L_{срОВ}$ – средняя длина обмотки возбуждения, м.

$$m_{ОВ} = 8900 \cdot 1,983 \cdot 10^{-5} \cdot 0,445 \cdot 287,3 = 22,628$$

5. Масса стали трубы статора, (кг):

$$m_j = 7800 \cdot L_{\delta_{полн}} \cdot S_{тр.ст},$$

$$m_j = 7800 \cdot 0,173 \cdot 0,012 = 16,517$$

6. Масса трубы ротора, (кг):

$$m_r = 7800 \cdot L_{\delta_{полн}} \cdot S_{тр.рот}$$

$$m_r = 7800 \cdot 0,173 \cdot 0,014 = 18,946$$

7. Масса магнитов, (кг):

$$m_{mag} = 7800 \cdot L_{\delta} \cdot \pi \left[\left(\frac{D_{рот} - 2h_{спинкирот}}{2} \right)^2 - \left[\frac{D_{рот} - 2(h_{спинкирот} + h_{маг})}{2} \right]^2 \right],$$

$$m_{mag} = 7800 \cdot 0,173 \cdot 3,14 \times$$

$$\left[\left(\frac{0,0243 - 2 \cdot 0,02}{2} \right)^2 - \left[\frac{0,0243 - (2 \cdot 0,02 + 0,011)}{2} \right]^2 \right] = 9,015$$

8. Полная масса активных материалов, (кг):

$$\sum m = m_z + m_a + m_{cu} + m_{ОВ} + m_j + m_r + m_{mag},$$

$$\sum m = 12,51 + 3,34 + 5,524 + 22,628 + 16,517 + 18,946 + 9,015 = 88,488$$

4.7 Расчет потерь в КПД

1. Электрические потери, (Вт):

$$P_{эл} = I_{ном}^2 \cdot r_{a75} \cdot (m - 1),$$

$$P_{эл} = 1,071 \cdot 10^{-3^2} \cdot 4,179 \cdot 10^{-4} \cdot (3 - 1) = 959,473$$

2. Электрические потери в обмотке возбуждения, (Вт):

$$P_{ОВ} = r_{ОВ} \cdot I_{ов}^2,$$

где $r_{ОВ}$ – сопротивления обмотки возбуждения, Ом.

$$P_{ОВ} = 0,141 \cdot 99,143^2 = 1,388 \cdot 10^3$$

4.7. Расчет потерь в зубцах

Входные параметры: $k_{dz} = 2$; $k_{да} = 1,5$; $\rho_{50} = 3,5$

1. Потери спинки, (Вт):

$$P_z = k_{dz} \cdot \rho_{50} \cdot B_{Z\text{середина}}^2 \left(\frac{f}{50}\right)^{1,3} \cdot m_z,$$

$$P_z = 2 \cdot 3,5 \cdot 0,858^2 \left(\frac{f}{50}\right)^{1,3} \cdot 12,518 = 269,349$$

2. Потери в спинке статора, (Вт):

$$P_a = k_{dz} \cdot \rho_{50} \cdot B_a^2 \left(\frac{f}{50}\right)^{1,3} \cdot m_a,$$

$$P_a = 2 \cdot 3,5 \cdot 0,759^2 \left(\frac{f}{50}\right)^{1,3} \cdot 3,34 = 56,193$$

3. Суммарные магнитные потери, (Вт):

$$P_{\text{маг}} = P_z + P_a$$

где P_z – потери спинки, (Вт).

$$P_{\text{маг}} = 269,349 + 56,193 = 325,542$$

4. Механические потери, (Вт):

$$P_{\text{мех}} = 0$$

5. Добавочные потери, (Вт):

$$P_{\text{доб}} = 0,005 \cdot I_{\text{ном}} \cdot U_n,$$

$$P_{\text{доб}} = 0,005 \cdot 1,071 \cdot 10^{-3} \cdot 28 = 150$$

6. Суммарные потери, (Вт):

$$\sum P = P_{\text{эл}} + P_{\text{ОВ}} + P_z + P_{\text{мех}} + P_{\text{доб}}$$

$$\sum P = 959,473 + 1,388 \cdot 10^3 + 269,349 + 0 + 150 = 2,823 \cdot 10^3$$

7. Потребляемая мощность, (Вт):

$$P_{\text{потреб}} = \frac{U_d \cdot I_d}{1},$$

$$P_{\text{потреб}} = \frac{U_d \cdot I_d}{1} = 3 \cdot 10^4$$

8. КПД:

$$\eta = \frac{P_{\text{потреб}}}{P_{\text{потреб}} + \sum P}, \text{ где } P_{\text{потреб}} \text{ – потребляемая мощность, Вт.}$$

$$\eta = \frac{3 \cdot 10^4}{3 \cdot 10^4 + 2,823 \cdot 10^3} = 0,914$$

9. Номинальный момент, (Н·м):

$$M_{\text{ном}} = \frac{P_{\text{потреб}} + \sum P}{2 \cdot \pi \cdot \frac{n}{60}},$$

$$M_{\text{ном}} = \frac{3 \cdot 10^4 + 2,823 \cdot 10^3}{2,3,14 \cdot \frac{3000}{60}} = 104,479$$

10. Постоянная двигателя:

$$C_E = \frac{N_{\text{пол}} \left(\frac{z \cdot u_n \cdot 2}{3} \right)}{\pi \cdot a}, \text{ где } u_n \text{ — число проводников у провода; } a \text{ — коэффициент}$$

полюсной дуги.

$$C_E = \frac{\frac{6 \left(\frac{18 \cdot 1,025 \cdot 10^3 \cdot 2}{3} \right)}{2}}{3,14 \cdot 1} = 31,315$$

$$M_n = C_E P_{\text{оток}} \cdot I_{\text{ном}} \cdot \frac{D}{D + h_{\text{паза}} \cdot 1},$$

$$M_n = 31,315 \cdot 8,932 \cdot 10^{-3} \cdot 1,071 \cdot 10^{-3} \cdot \frac{0,243}{0,243 + 0,021 \cdot 1} = 272,958$$

10. Ток короткого замыкания, (А):

$$I_{kz} = \frac{20}{(m-1) \cdot r_{a75}}$$

$$I_{kz} = \frac{20}{(3-1) \cdot 4,179 \cdot 10^{-4}} = 2,393 \cdot 10^4$$

11. Момент короткого замыкания (Н·м):

$$M_{kz} = C_E \cdot I_{kz} \cdot I_{kz} \cdot \frac{D}{D + h_{\text{паза}} \cdot 1},$$

$$M_{kz} = 31,315 \cdot 8,932 \cdot 10^{-3} \cdot 4,179 \cdot \frac{0,243}{0,243 + 4,179 \cdot 1} = 6,096 \cdot 10^3$$

Завершение расчета.

На рисунке 26 изображена внутренняя геометрия ВЭКВ.

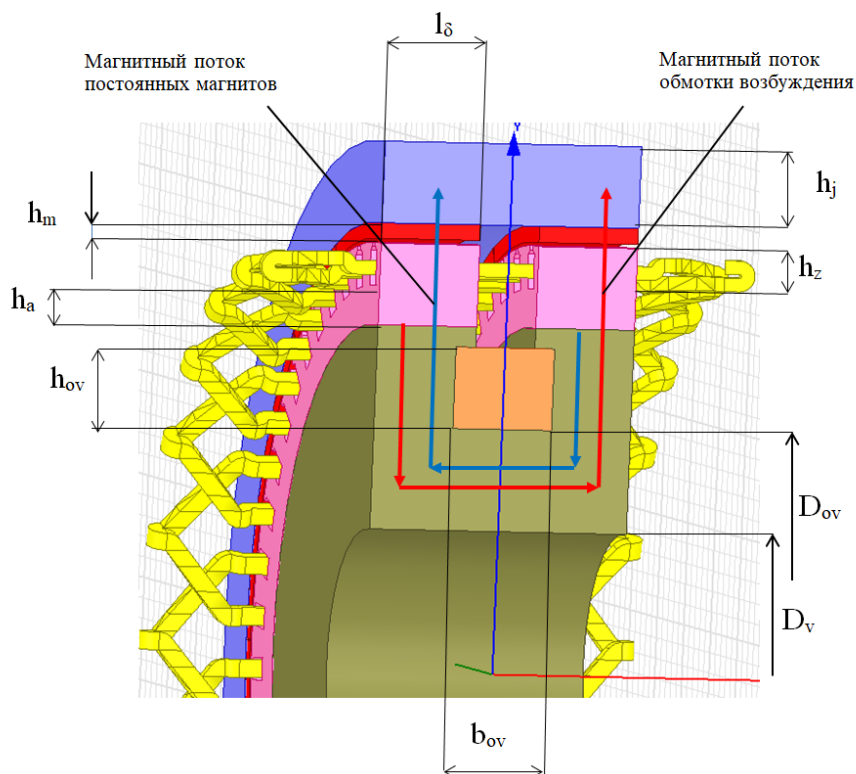


Рисунок 26 – Геометрия ВЭКВ

Вывод по четвертой главе

Были построены графики предсказанных коэффициентов для трех параметров. Было проведено сравнение двух разных моделей нейронной сети: многослойного перцептрона и рекуррентной нейронной сети. Была протестирована функция предсказания коэффициентов. Была рассчитана внутренняя геометрия ВЭКВ.

ЗАКЛЮЧЕНИЕ

В рамках данной работы была разработана система для оптимизации параметров вентильного двигателя постоянного тока используя возможности нейросетей с помощью многослойного перцептрона и рекуррентной нейронной сети. При это были решены следующие задачи:

1) выполнен анализ предметной области, выполнен обзор научной литературы;

2) выполнен анализ требований к проектируемой системе;

3) разработаны модели на основе нейросетей;

4) проанализирована эффективность моделей и было проведено сравнение моделей. Были использованы полученные данные обобщенных переменных для инженерной методики проектирования ВЭКВ.

Полученные результаты могут стать основой для дальнейшего исследования эффективности применения различных методов машинного обучения для решения рассматриваемой проблемы.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Рекуррентные нейронные сети (RNN) с Keras. [Электронный ресурс] URL: <https://habr.com/ru/articles/487808/> (дата обращения: 02.10.2024 г.)
2. How does a neural network work. [Электронный ресурс] URL: <https://mize.tech/blog/how-does-a-neural-network-work-implementation-and-5-examples/> (дата обращения: 02.11.2024 г.)
3. Предсказание временных рядов с помощью Keras. [Электронный ресурс] URL: <https://habr.com/ru/articles/677714/> (дата обращения: 02.10.2024 г.)
4. TensorFlow documentation. [Электронный ресурс] URL: https://www.tensorflow.org/api_docs/python/tf/all_symbols (дата обращения: 03.02.2024 г.)
5. Sklearn documentation. [Электронный ресурс] URL: <https://scikit-learn.org/0.21/documentation.html> (дата обращения: 03.02.2024 г.)
6. NumPy documentation. [Электронный ресурс] URL: <https://numpy.org/doc/> (дата обращения: 03.02.2024 г.)
7. Pandas documentation. [Электронный ресурс] URL: <https://pandas.pydata.org/docs/> (дата обращения: 03.02.2024 г.)
8. Matplotlib 3.9.0 documentation. [Электронный ресурс] URL: <https://matplotlib.org/stable/index.html> (дата обращения: 03.02.2024 г.)
9. SciPy documentation. [Электронный ресурс] URL: <https://docs.scipy.org/doc/scipy/> (дата обращения: 04.05.2024 г.).
10. Google Colaboratory [Электронный ресурс]. URL: <https://colab.research.google.com/notebooks/welcome.ipynb?hl=ru> (дата обращения: 10.03.2024 г.).
11. Keras [Электронный ресурс]. URL: <https://keras.io/> (дата обращения: 10.03.2024 г.).
12. Вигерс Карл. Разработка требований к программному обеспечению. // Москва: Издательско-торговый дом Русская редакция, 2004. – 576 с.
13. Старолетов С.М. Основы тестирования и верификации программного обеспечения. // Санкт-Петербург: Изд-во Лань, 2020. – 344 с.

14. What are Brushless DC Motors. [Электронный ресурс]. URL: <https://www.renesas.com/us/en/support/engineer-school/brushless-dc-motor-01-overview> (дата обращения: 10.02.2024 г.).

15. Рекуррентная нейронная сеть (RNN): виды, обучение, примеры. [Электронный ресурс]. URL: <https://neurohive.io/ru/osnovy-data-science/rekurrentnye-nejronnye-seti/> (дата обращения: 15.03.2024 г.).

16. Пат. 2015606 Российская Федерация, МПК Н 02 К 5/22. Электродвигатель / Ганджа С. А.; заявитель и патентообладатель С. А. Ганджа. - № 5039128/07; заявл. 21.04.92; опубл. 30.06.94.

17. Gandzha S., Aminov D., Bakhtiyor K. Application of the combined excitation submersible hydrogenerator as an alternative energy source for small and medium rivers. IEEE Russian Workshop on Power Engineering and Automation of Metallurgy Industry. 4-5 Oct. 2019 Magnitogorsk, Russia. DOI: [10.1109 / PEAMI.2019.8915294](https://doi.org/10.1109/PEAMI.2019.8915294).

18. Евгеньев, Г.Б. Case-технология создания многоагентных САПР изделий машиностроения/ Евгеньев Г.Б. [и др.] // Труды Международных научно-технических конференций «Интеллектуальные системы» и «Интеллектуальные САПР». - Т. 2. -М.: Физматлит, 2003. - С. 41-46.

19. Иванов-Смоленский А. В. Электрические машины: в 2 т. Т. 2. / А. В. Иванов-Смоленский. — 2-е изд., перераб. и доп. - М.: Изд-во МЭИ, 2005. - 528 с.

20. [Кондратьев, В. А.](#) Статические и динамические силы электромагнитных механизмов/ В. А. [Кондратьев](#), В. Л. Малинин // Транспорт: наука, техника, управление. - 2008. - №6. - С.24-26.

21. Копылов, И. П. Некоторые проблемы создания автоматизированных систем проектирования электрических машин // Всесоюзная конференция «Современные проблемы энергетики и электротехники»: тез. докл. - М., 1977.

22. Копылов И.П. Электрические машины / И. П. Копылов. — 5-е изд. - М.: Высш. шк., 2006. - 607 с.

23. Уайлд, Д. Д. Методы поиска экстремума: [пер с англ.]. - М.: Наука, 1967. – 268 с.

24. Проектирование электрических машин / И. П. Копылов, Б. К. Клоков, В. П. Морозкин, Б. Ф. Токарев. - 4-е изд. / под ред. И.П. Копылова. - М.: Высш. шк., 2005. – 767 с.