

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ И ПОСТАНОВКА ЗАДАЧИ	4
1.1 Основные понятия предметной области	4
1.2 Результаты обследования пастеризационной установки	5
1.3 Введение в прогнозирование, нейронные сети и машинное обучение	7
1.4 Требования к модулю прогнозирования и постановка задачи	12
2. АНАЛИЗ СУЩЕСТВУЮЩИХ РЕШЕНИЙ	14
Выполненные в ходе практики задачи по дипломному проекту	16
Выполненные индивидуальной задачи	18

						Лист
Изм.	Лист.	№ докум.	Подп.	Дата		2

ВВЕДЕНИЕ

На современном этапе развития человечества довольно сложно представить мир без нейронных сетей, которые продолжают развиваться и осваивать всё новые сферы и области человеческой деятельности. Одной из таких хорошо освоенных нейронными сетями сфер является производственная сфера. Обученные нейронные сети способны классифицировать продукты производства и определять их дальнейшую судьбу, способны предсказывать поведение некоторой системы для дальнейшего планирования производства, искать аномалии в данных технологического процесса и многое другое.

Данная работа как раз и посвящена прогнозированию данных технологического процесса производственного оборудования, а именно пастеризационной установки. Для выполнения данной задачи будут использоваться нейронные сети, поскольку решение задачи прогнозирования именно нейросетевым способом на сегодняшний день является одним из самых актуальных, эффективных, точных и изученных способов решения данной задачи. Мало того, что такой способ позволяет гибко адаптироваться к процессам и научиться достаточно точно их прогнозировать, так ещё и такой способ предусматривает возможность, так скажем, подстраивания, благодаря чему, можно улавливать новые особенности процесса. К тому же, существует большое количество различной литературы про прогнозирование с использованием нейронных сетей, что даёт уверенность в том, что такой способ обеспечивает необходимыми результатами, а также что можно достичь хороших прогнозов, используя опыт различных авторов.

По большей части, данная работа является смесью задачи прогнозирования и машинного обучения, поскольку в её рамках будет рассматриваться изучение предметной области и определение задач и целей прогнозирования, сбор, анализ и обработка данных технологического процесса пастеризационной установки, их подготовка к прогнозированию и обучению модели нейронной сети, после проектирование и реализация самой нейронной сети, её обучение, тестирование и оценка, а в заключительном этапе готовую модель необходимо развернуть для взаимодействия с пастеризационной установкой.

Дойдя до этапа создания модели, будут выбраны основные архитектуры нейронных сетей, подробно описаны их способы функционирования и обучения, а также детально будут рассмотрены результаты работы моделей и их сравнение между собой.

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ И ПОСТАНОВКА ЗАДАЧИ

1.1 Основные понятия предметной области

Объектом исследования и внедрения нейронной сети является пастеризационная установка. Для понимания её технологического процесса, необходимо для начала понять с чем она работает и что она производит, а также какими функциями обладает пастеризационная установка. Поэтому в данной главе мы будем рассматривать пастеризационную установку как некий «серый ящик»: опишем объекты, которые попадают на вход и выходят из пастеризационной установки, а также опишем самые основные процессы, которые происходят с подаваемым на вход объектом.

И так, основной объект обработки пастеризационной установки является молоко. Поэтому в дальнейшем молоко будет рассматриваться как объект технической обработки. Рассматривая его таким образом мы понимаем, что оно должно обладать некоторыми показателями, например, состав молока, степень чистоты, кислотность, наличие токсичных и нейтрализующих веществ. При этом молоко обладает ещё и различными свойствами: органолептическими, физико-механическими и биохимическими. И так, разберём лишь те параметры и свойства, которые будут нам в дальнейшем интересны.

Молоко можно разделить на две составляющих: вода и распределённые в этой воде пищевые вещества. К таким веществам относят жиры, белки, углеводы, ферменты, различные минеральные вещества и газы. Помимо этого в молоке могут находиться различные микроорганизмы. И как известно, некоторые из этих микроорганизмов, содержащиеся в молоке, являются опасными или вредными, например, бруцеллеза, ящура, возбудитель кишечной палочки и другие. Но как избавиться от вредных и опасных микроорганизмов? Для этого используется процесс пастеризации – уничтожение различных форм вредных и опасных микроорганизмов в молоке. Но при этом молоко должно сохранить свою биологическую и питательную ценность, а также и своё качество.

Однако перед тем, как перейти к пастеризации, необходимо определить, какое молоко можно пастеризовать, каким требованиям оно должно соответствовать. Для определения подходящего для пастеризации молока существует множество различных показателей и требований. Так, например, пригодное для пастеризации молоко должно быть кислотностью не более 22 °Т, а бактериальная обсеменённость молока должна быть один миллион клеток на сантиметр кубический. При этом, молоко не должно быть вспененным. Перед пастеризацией, молоко должно быть также предварительно очищено на фильтрах или на сепараторах-молокоочистителях. Что ж, основные требования перечислены.

А теперь, к самой пастеризации. Основные параметры пастеризации есть температура пастеризации, а также время выдержки, т.е. время нахождения молока в данном процессе. Относительно данных параметров существует формула пастеризации, выведенная Г. А. Куком и называемая критерием Пастера:

$$P = \frac{t}{\tau}$$

где, t – время действия температуры пастеризации, с; τ – время бактерицидного действия температуры пастеризации, с.

Что такое бактерицидного действие температуры пастеризации? Это, как раз и есть эффект, в результате которого происходит уничтожение вредных и опасных микроорганиз-

						Лист
Изм.	Лист.	№ докум.	Подп.	Дата		4

мов.

Также известна и ещё одна немаловажная зависимость: продолжительность выдержки зависит от температуры пастеризации. Это можно увидеть в следующем выражении:

$$\ln t = 36.84 - 0.48T$$

где, T – температура пастеризации, °C.

Завершение процесса пастеризации характеризуется полным уничтожением содержащихся в молоке микроорганизмов. Это можно будет определить благодаря уже известному критерию Пастера, значение которого должно быть не меньше единицы, для того, чтобы считать, что процесс пастеризации завершён.

Перед тем, как перейти к описанию пастеризационной установки, следует коротко разобрать ещё одно понятие, с которым мы будем сталкиваться в дальнейшем, а именно гомогенизация. Она представляет собой процесс дробления или уничтожения жировых шариков, образовавшихся в ходе хранения молока. Под воздействием внешних сил можно достичь значительного уменьшения объёма жировых шариков. Процесс гомогенизации позволяет предотвратить самопроизвольное отстаивание жира в молоке на производстве или при его хранении. При этом, гомогенизация даёт возможность сохранить однородную консистенцию молока. Далее рассматривать гомогенизацию так подробно, как процесс пастеризации, не имеет смысла, поскольку процесс гомогенизации не является центральным понятием предметной области.

1.2 Результаты обследования пастеризационной установки

Пастеризационная установка, она же пастеризационно-охладительная установка, предназначена непосредственно для самой пастеризации, т.е. для уничтожения вредных и опасных микроорганизмов в молоке. Она состоит из множества различных элементов, изучая которые мы также сможем изучить принцип работы пастеризационной установки.

Строго говоря, пастеризационная установка состоит из:

1. Балансного танка
2. Подающего насоса
3. Регулятора потока
4. Секции регенеративного предварительного подогрева
5. Центробежного очистителя
6. Секции нагрева
7. Трубы выдержки
8. Вспомогательного насоса
9. Системы нагрева горячей воды
10. Секции регенеративного охлаждения
11. Секции охлаждения
12. Возвратного клапана
13. Панели управления

И так, балансный танк является по сути резервуаром с молоком, оборудованным поплавковым входным клапаном, который регулирует расход молока и поддерживает его постоянный уровень в резервуаре. В балансном танке также имеется электрод минимального уровня. Он срабатывает, в том случае, когда уровень молока достигает минимальной

						Лист
Изм.	Лист	№ докум.	Подп.	Дата		5

точки, тем самым включая клапан распределения потока. Молоко заменяется водой и пастеризатор отключается.

Подающий насос позволяет обеспечить молоком сам пастеризатор, выкачивая его из балансного танка.

Для обеспечения устойчивого контроля температуры и постоянного времени выдержки пастеризационной установки используется регулятор потока. Он также необходим для поддержки расхода через пастеризатор на должно уровне.

Молоко, попав в секцию предварительного регенеративного подогрева, должно приобрести некоторую начальную температуру. Это осуществляется с помощью регенерированного тепла уже пастеризованного молока, которое в этот момент должно охлаждаться в секции регенеративного охлаждения.

Секцию регенеративного нагрева можно разделить на три секции: секция начального регенеративного подогрева, секция предварительного очищения и секция конечного регенеративного подогрева. Такое разделение используется, поскольку существует необходимость в предварительной обработке молока между входной и выходной температурой молока в секции регенеративного подогрева. Секция начального регенеративного подогрева обычно придаёт молоку температуру около 55 °С. После молоко отправляется в секцию предварительного очищения, представленную центробежным очистителем. После молоко поступает в секцию конечного регенеративного подогрева, где молоко добывает максимальную температуру.

Эффективность такого нагревания молока регенеративным способом достигает 90-96%, что отлично сказывается на энергосбережении. Остальную температуру входное молоко добывает за счёт горячей воды, температура которой на 2 или 3 °С больше температуры пастеризации. Все перечисленные процессы происходят в секции подогрева. Горячая вода имеется благодаря системе нагрева воды.

Нагретое молоко попадает во внешнюю трубу выдержки, где температура проверяется датчиком. Этот датчик непрерывно контактирует с регулятором температуры на панели управления, а также воздействует на регистрирующий прибор, который сохраняет температуру пастеризации. Тем самым происходит контроль процесса пастеризации.

Как только температура падает ниже минимума, активизируется возвратный клапан, позволяющий пастеризованному молоку попасть в секцию регенеративного охлаждения.

Секцию регенеративного охлаждения также можно разделить на две секции: первая секция регенеративного охлаждения и вторая секция регенеративного охлаждения. В первой секции регенеративного охлаждения горячее пастеризованное молоко передаёт свою температуру холодному, предварительно очищенному молоку, находящегося в секции регенеративного конечного подогрева молока. А во второй секции регенеративного охлаждения пастеризованное молоко отдаёт температуру необработанному молоку, находящемуся в секции регенеративного начального подогрева.

После охлаждённое молоко попадает в секцию охлаждения, которую тоже можно разделить на две секции: первая секция охлаждения и вторая секция охлаждения. В первой секции пастеризованное молоко охлаждается холодной водой, а во второй секции молоко уже охлаждается ледёной водой.

						Лист
						6
Изм.	Лист.	№ докум.	Подп.	Дата		

1.3 Введение в прогнозирование, нейронные сети и машинное обучение

А сейчас опишем средства, которые будут использоваться для составления прогнозов, а также опишем само понятие прогноза.

Одним из ключевых понятий данной работы является понятие временных рядов. Что же это? Временные ряды – это, по сути, некоторая последовательность, каждый элемент из которой состоит из двух или более параметров, один из которых обязательно должен обозначать время. Причём, все эти элементы в последовательности расположены в хронологическом порядке, т.е. в порядке возрастания параметра времени.

Параметр времени может быть представлен в разных форматах. Выбор формата времени зависит от задачи, удобства использования, длительности, в пределах которой будут собираться данные, а также от требуемой точности. Например, в случае, если данные фиксируются раз в день, то хорошо подойдёт отсчёт времени по дням с указанием месяца и года. Если же данные фиксируются в определённые моменты дня, то к вышеописанному стоит прибавить указание часа и минуты фиксации. При необходимости можно указывать и секунды, и миллисекунды. Но что, если нам не особо-то и важно знать, в какой год, месяц или день это происходило, когда нам важно знать, сколько прошло времени от начала того или иного процесса? Тогда, нам скорее подойдёт формат дискретного времени. С помощью этого формата можно узнать длительность процесса в единственной выбранной нами единице измерения времени. Например, если мы сохраняем время в секундах, то 1000-ча секунд сохранит свой формат 1000-чи секунд, время не будет переведено в 16-ать минут и 40-ок секунд. Всё это нам позволяет не привязываться к датам, которые не особо-то и влияют на технологический процесс промышленного оборудования.

Остальные параметры могут уже характеризовать или описывать какой-либо процесс или процессы, причём даже не обязательно одного элемента, а даже целой системы элементов. Так, например, когда наш элемент последовательности состоит из двух признаков, а один из которых, как мы уже знаем, время, то второй, конечно же, уже будет обозначать характеристику или состояние изучаемого нами элемента. Но как только у нас появляется три или более признаков, тогда мы уже можем говорить о фиксации характеристик или состояний разных элементов изучаемой системы в один и тот же момент времени или же о фиксации характеристики или состояния одного из множества элементов системы, но с указанием этого элемента, например, с помощью идентификационного номера элемента. Как можно убедиться, временные ряды дают весьма гибкую возможность описания процессов или систем относительно времени.

Технологический процесс пастеризационной установки как раз и представляет собой временной ряд, в котором имеется информация о показаниях различных датчиков в определённые моменты времени. Поэтому, говоря о данных технологического процесса пастеризационной установки, мы будем понимать, что они имеют форму временных рядов.

А что из себя представляет работа с временными рядами? В основном работа делится на две части. Первая часть – это понимание структуры временного ряда, его закономерностей, таких как цикличность, тренд, сезонность и так далее, обработка данных временного ряда, визуализация данных, в общем, это всесторонний анализ временного ряда. Если опускать различные математические, статистические и тому подобные подробности, то анализ также может дать нам возможность понять, как начинался процесс, как шёл и развивался и на чём он закончился или остановился на данном моменте времени. Стро-

го говоря, нейронным сетям, как и исследователям временных рядов, тоже необходимо это понять, чтобы выполнить вторую часть работы, а именно, составление прогноза, что, зачастую, и является основной задачей работы с временным рядом. Да, анализ данных временного ряда технологического процесса делается с целью понять, что будет происходить с этим процессом дальше. Но чем нам так полезна информация о будущем, зачем она нужна? Для этого необходимо понять, что есть прогноз.

Сам прогноз – это некоторая случайная величина, характеризующая вероятность того, что график в будущем пройдёт через определённую точку или некоторую область. Тогда прогнозирование – это получение максимально точных прогнозов, или, если говорить в отрыве от понятия прогноза, то это точное предсказание будущего, учитывающее исторические данные об объекте прогнозирования, а также знания о любых будущих событиях, которые могут повлиять на прогнозы.

Как понять, что прогнозы действительные? Прогнозы являются таковыми, если они отражают подлинные закономерности и взаимосвязи, которые есть в исторических данных, при этом не повторяя прошлые события, которые более не актуальны или уже не повторяются.

Что необходимо, чтобы составить хороший прогноз на основе временных рядов? Для этого обычно требуется выполнить следующие шаги:

1. Определить задачу.
2. Собрать информацию.
3. Произвести предварительный анализ.
4. Выбрать и создать модель прогнозирования.
5. Использование и оценивание модели прогнозирования.

Вкратце разберём каждый пункт. При определении задачи необходимо понять что вообще будет прогнозироваться, как будут использоваться прогнозы, благодаря чему будут получены прогнозы, кому эти прогнозы нужны или для чего. Второй пункт подразумевает непосредственный сбор или получение данных, а также оценка накопленного опыта людей, которые собирают данные и используют прогнозы. Для выполнения третьего пункта необходимо визуализировать данные, составить инфографику, если она необходима, определить взаимосвязь с признаками, закономерности временных рядов, качество данных и так далее, другими словами, провести анализ данных. На четвёртом пункте необходимо либо приобрести и адаптировать готовую модель, либо создать её самостоятельно. Но модель должна учитывать исторические данные, силы взаимосвязи прогнозируемым признаком и любым другим признаком, а также способы использования прогнозов. В заключении проводится тестирование, оценивание, развёртывание и сопровождение модели прогнозирования.

В дальнейшем на эти пять шагов прогнозирования мы будем ссылаться как на пять этапов составления прогноза.

Теперь поговорим о нейронных сетях и машинном обучении, а также о том, какое место занимает задача прогнозирования в рамках машинного обучения.

О нейронных сетях мы будем говорить, как о некоторой математической модели и её программной реализации, которая имитирует работу и взаимосвязи настоящих нейронов. Рассматривая нейронную сеть как «чёрный ящик», изображённую на рисунке 1.1, мы понимаем, что есть некоторое n входных данных и m преобразованных выходных данных. В зависимости от назначения сети входные данные могут обозначать исторические данные о процессе, необработанные картинки, зашумлённые чёрно-белые фотографии, текст с ошибками и так далее, а выходными данными могут быть прогнозы, картинки с разме-

ченными на них объектами, восстановленные цветные фотографии, исправленные тексты и так далее. Как можно понять, нейронные сети весьма универсальное средство, которое может решать большой спектр задач.

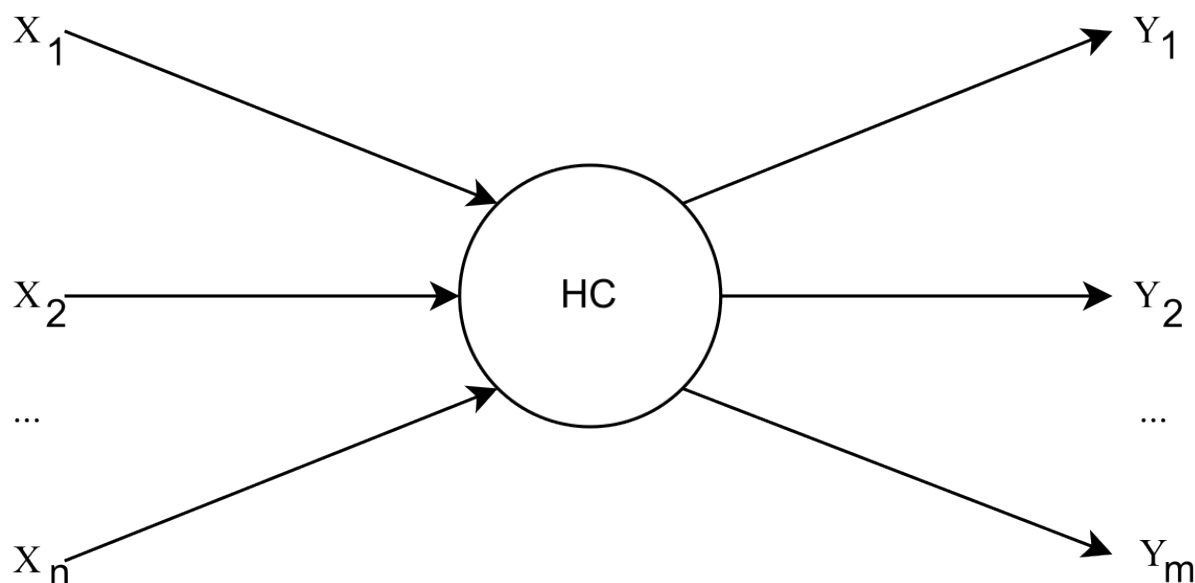


Рисунок 1.1: Представление нейронной сети как "чёрный ящик"

Нейронная сеть, если её рассматривать уже как «белый ящик», состоит из нейронов. Эти нейроны могут связываться между собой различными способами, например, один нейрон к одному нейрону, один ко многим, иметь обратные связи и так далее. По большей части, то, какие связи имеются между нейронами в сети, определяет, так называемую, архитектуру нейронной сети. Далее нейроны собираются в упорядоченные нейронные слои. Таких слоёв может быть несколько. Тот слой, первый из которых работает с входными данными, называется входным слоем. Слой, который производит расчёт выходных значений, называется выходным слоем. Остальные слои называются скрытыми. Каждый из этих слоёв имеет собственную размерность. На рисунке 1.2 нейронные слои изображены как вертикальные последовательности из нейронов, изображённые кружочками.

Слои, как уже было сказано, состоят из нейронов. Каждый нейрон, в свою очередь, состоит из сумматора и активирующей функции. В первом элементе нейрона может происходить комбинирование операции умножения, сложение, умножения Адамара и так далее. Как правило, вначале производится умножение между входными данными нейрона и весами, каждый из которых соответствует своему входу. Веса могут обозначаться следующими символами: W , w , V , v , U , u и т.п. Веса могут быть представлены в виде отдельных значений, либо векторов, но чаще всего, в виде матриц. Результат каждого умножения в дальнейшем складываются между собой, а после выполняется операция сложения со смещением, которое обычно обозначается символами B или b , или операция вычитания порога, который обычно обозначается символами T или t . Как правило, смещения и пороги представлены в виде одного значения, либо некоторого вектора, размерность которого должна равняться размерности выхода нейрона. Результат всех этих операций в конечном итоге преобразуется с помощью активирующей функции, в качестве которой могут выступать такие функции как: линейная, тождественная, логистическая (сигмоидальная), биполярная логистическая, функции гиперболического тангенса, ReLU и так далее.

Пример расчёта нейрона:

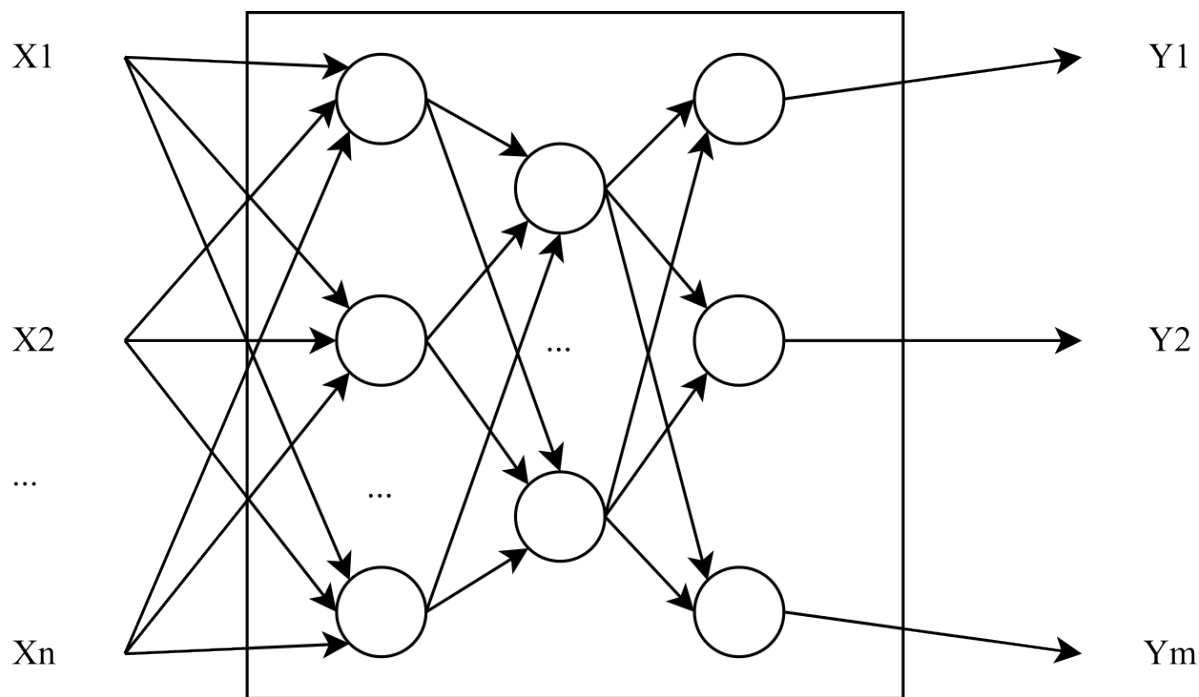


Рисунок 1.2: Представление нейронной сети как "белый ящик"

$$y = F_{act} (w_1x_1 + w_2x_2 + \dots + w_nx_n + b)$$

И так, теперь перейдём к машинному обучению – набор алгоритмов, благодаря которым осуществляется обучение нейронной сети. А её обучение можно ещё определить, как подбор или поиск таких значений весов и смещений или порогов каждого нейрона, чтобы нейронная сеть выдавала необходимые результаты.

Теперь давайте определим, как связана задача прогнозирования и машинное обучение, чтобы обучить нейронную сеть предсказывать будущее. Чтобы это понять, нужно иметь в виду, что машинное обучение делится на четыре направления: обучение с учителем, обучение без учителя, обучение с подкреплением и обучение с частичным привлечением учителя. Разберём только обучение с учителем и обучение без учителя, поскольку другие виды обучения не будут затронуты вообще.

Однако, для начала необходимо разъяснить несколько понятий, которые используются в лексике машинного обучения. Первое из таких понятий это понятие признака – некоторое измерение или параметр, определяющий характеристику или состояние чего-либо. Следующее понятие есть метка – по сути то, чему должен соответствовать признак. И в конце концов, пример – это то, что подаётся на вход нейронной сети. Существуют помеченные и непомеченные примеры. Помеченные примеры — это такие примеры, которые состоят из пар признаков и меток, т.е. $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$, или такие примеры, в которых каждому вектору или каждой последовательности признаков соответствует вектор или последовательность меток $\{(X_1, Y_1), (X_2, Y_2), \dots, (X_n, Y_n)\}$. Непомеченные примеры, это по факту просто признаки, т.е. $\{x_1, x_2, \dots, x_n\}$.

И так, говоря об отличии между обучением с учителем и обучением без учителя, можно сказать, что обучение с учителем использует помеченные примеры, а обучение без учителя использует непомеченные примеры.

Рассмотрим задачу прогнозирования и определим, какими примерами мы оперируем, чтобы выбрать подходящую модель обучения модели нейронной сети. И так, для

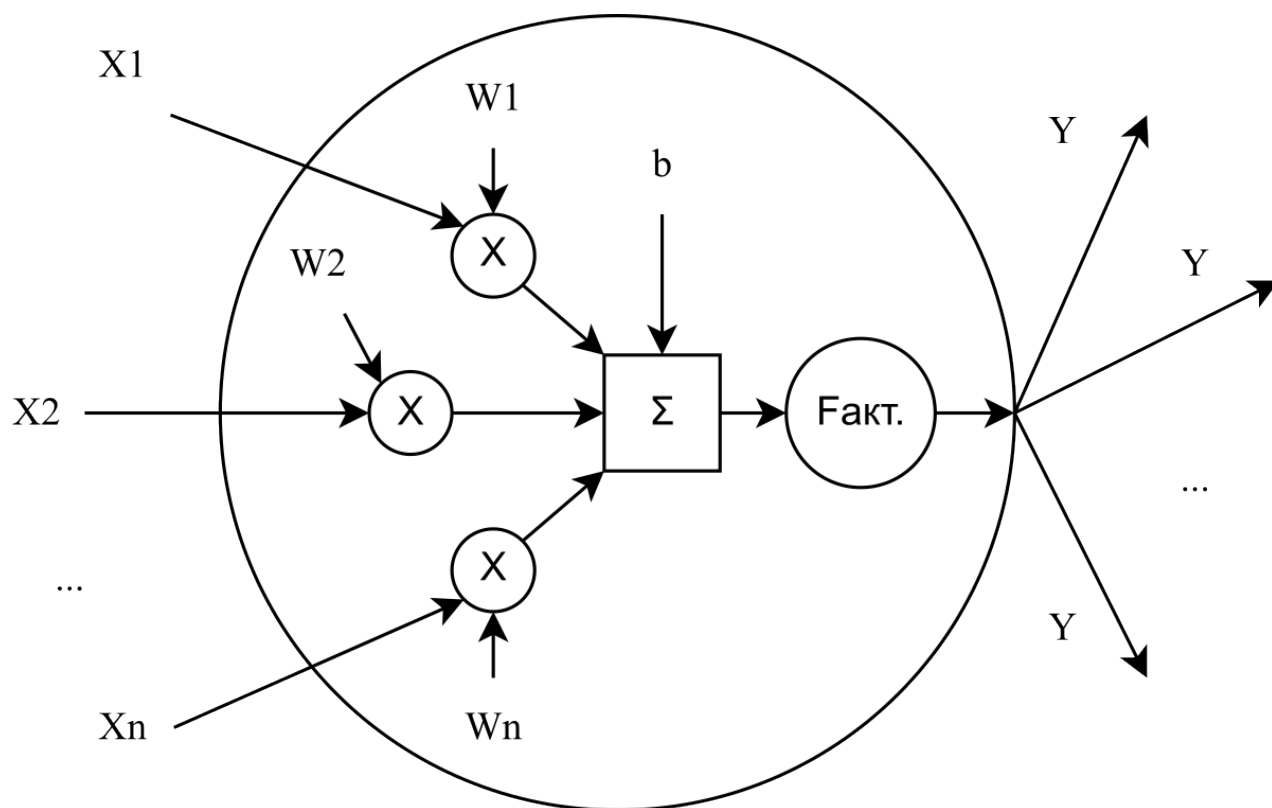


Рисунок 1.3: Представление нейронной сети как "белый ящик"

прогнозирования используются исторические данные, а данных из будущего у нас нет. В таком случае, как бы мы не рассчитывали будущее с помощью нейронных сетей на основе исторических данных без проверки с тем, что должно получиться, мы не достигнем никаких результатов. Нам необходимы некоторые эталоны, с которыми мы могли сравнивать полученные результаты сети. Откуда их брать? Чтобы их получить мы разделим исторические данные на примеры и соответствующие им эталоны, т.е. метки. Таким образом для обучения сети мы получим помеченные данные, в которых каждому вектору признаков, взятого из исторических данных, будет соответствовать вектор меток, взятых из тех же исторических данных, взятых сразу после последнего значения вектора признаков из общей последовательности исторических данных. Тем самым, мы говорим, что наша задача прогнозирования может быть решена с помощью нейронных сетей и машинного обучения.

Раз мы определились, что мы будем использовать машинное обучение для решения задачи прогнозирования, стоит пару слов сказать о инженерии машинного обучения, которая поможет достичь нам необходимых результатов. Рассмотрим этапы инженерии машинного обучения, которые определяют последовательность дальнейшей работы:

1. Определение задачи и цели.
2. Сбор и подготовка данных.
3. Конструирование признаков.
4. Обучение модели.
5. Оценка модели.
6. Развёртывание модели.
7. Обслуживание запросов к модели.
8. Мониторинг модели.
9. Сопровождение модели.

Первый, второй и третий этапы инженерии машинного обучения схожи с первы-

ми тремя этапами составления прогноза. На эта обучения модели произойдет непосредственно само обучение сети, по правилам, в соответствии с типом обучения. На пятом этапе проводится всесторонняя оценка модели, а также настройка некоторых её элементов. Именно на этом этапе делается вывод о готовности модели нейронной сети выполнять поставленные задачи. После наступает этап развёртывания нейронной сети, в ходе которого модель нейронной сети попадает в целевую среду для выполнения поставленных перед ней задач. На этапе обслуживания запросов необходимо убедиться в том, что на входы нейронной сети будут попадать данные такого формата, который будет понятен модели, и что сеть выдаёт необходимые результаты. Другими словами это вопросы снабжения модели сети всем необходимым для её функционирования. Далее идёт этап мониторинга нейронной сети и её сопровождения. По факту это этапы использования сети по назначению.

Изучив этапы инженерии машинного обучения, становится понятно, что необходимо сделать, чтобы была создана модель нейронной сети, способная давать приемлимые результаты. Поэтому дальнейшая работа будет развиваться относительно этапов инженерии машинного обучения.

И так, порядок работы определён, поэтому в следующей главе будет разберано, какие требования предъявляются к самой модели для прогнозирования, после будут определены цели и задачи прогнозирования, выполнив тем самым первые этапы инженерии машинного обучения и построения прогноза.

1.4 Требования к модулю прогнозирования и постановка задачи

Модуль прогнозирования должен выполнять прогноз технологического процесса пастеризационной установки. Для этого ему необходимо получать данные технологического процесса пастеризационной установки с помощью программируемого микроконтроллера, использующего платформу PLCnext Technology. По этой причине, программа должна быть написана на низкоуровневом языке программирования, а именно на C++, который поддерживается данным микроконтроллером.

Поскольку процесс пастеризации происходит непрерывно, то программа должна работать в режиме реального времени и своевременно снабжать необходимой информацией о прогнозах поведения пастеризационной установки.

Программа должно быть устойчива к различным сбоям и авариям, а также не создавать их сама.

Должна быть предоставлена возможность вносить некоторые настройки в модуль прогнозирования для корректировки работы модели нейронной сети.

Для постановки задачи, необходимо понимать, что сама программа, которая будет в дальнейшем использоваться для прогнозирования данных временных рядов технологического процесса пастеризационной установки, будет использовать нейросетевое решение задачи прогнозирования. Для обучения модели нейронной сети применяются алгоритмы машинного обучения, от которых зависит качество модели прогнозирования. Оттого данная работа в основном и будет развиваться по этапам инженерии машинного обучения, и потому данная работа будет являться проектом машинного обучения.

Задачей для проекта машинного обучения будет является разработка модуля прогнозирования данных временных рядов пастеризационной установки с целью прогноза поведения пастеризационной установки, планирования производства, а также выявления

аномального поведения, возможных сбоев и выбросов пастеризационной установки.

Для выполнения поставленной задачи необходимо изучать данные технологического процесса, а также провести их анализ и обработку. Далее необходимо выбрать подходящую архитектуру нейронной сети, построить и обучить модель сети, после чего организовать эффективное обучение сети, используя алгоритмы машинного обучения. Написать код нейронной сети необходимо на низкоуровневом языке программирования для дальнейшего развёртывания на программируемом контроллере, взаимодействующим с пастеризационной установкой.

Также необходимо произвести тестирование и оценку качества полученной модели. После чего непосредственно выполнить процесс развёртывания нейронной сети.

						Лист
Изм.	Лист.	№ докум.	Подп.	Дата		13

2. АНАЛИЗ СУЩЕСТВУЮЩИХ РЕШЕНИЙ

В этой главе будут проанализированы методы, которые были использованы при решении схожей задачи в организации Kaspersy.

Перед организацией ставилась цель разработать систему, которая позволяла бы предотвращать отказы, аварии и незапланированные простои промышленного оборудования, выявляя признаки проблемы задолго до того, как проблема повлияет на работу предприятия. Другими словами, необходимо произвести прогнозирование данных технологического процесса промышленного оборудования для выявления признаков, по которым можно говорить о будущих аномалиях. Под аномалией для Kaspersky будет пониматься значительное отклонение реальных данных технологического процесса от прогноза.

Причинами, по которым Kaspersky занялся данной задачей, стали устаревшие системы мониторинга, не способные эффективно искать причины возникновения аварий, а также огромный объем телеметрии, с которыми не могли справиться даже опытные операторы.

Для решения таких проблем была создана система Kaspersky MLAD. Это система для наблюдения за большим количеством пользователей телеметрии, а также для выявления отклонений, которые могут нанести повреждения производству.

Данная система способна предотвращать неполадки различного рода, представленные дефектом или сбоем оборудования, или ошибкой персонала, с целью дальнейшего предотвращения развития опасной ситуации. Также система способна определять отклонения в действиях сотрудников, позволяя раскрыть бойкот, диверсию, срыв производства и так далее. В добавок система Kaspersky MLAD призвана выявлять атаки злоумышленников.

Для работы системы Kaspersky MLAD не требуется изменять как-либо технологической процесс промышленного оборудования. Эта система никак не воздействует на него. К тому же она не вмешивается в передачу данных или в средства управления промышленным оборудованием. Данная система оповещает оператора о возможном сбое, и оставляет задачу принятия решения на операторе. Данным свойством будет обладать и модели, описанные в данной работе.

Если рассматривать Kaspersky MLAD с технической стороны, то это программное обеспечение, которое использует нейронные сети, способные анализировать существующий поток телеметрии. Поскольку параметры телеметрии связаны друг с другом и влияют друг на друга, то система Kaspersky MLAD должна это учитывать, и она с этим справляется. Она способна выявлять взаимосвязи между параметрами и использовать их для построения прогноза. Этот прогноз в дальнейшем обрабатывается различными нейросемантическими средствами, выявляя тем самым аномалии.

Система Kaspersky MLAD использует различные архитектуры нейронных сетей, такие как: DenseNet, CNN, TCN и RNN. Первые три сети используются для различного рода обработки данных, их анализа и выявление аномалий. А сети RNN используются для прогнозирования данных технологического процесса промышленного оборудования. Данная работа последует примеру Kaspersky MLAD и также будет использовать RNN сети для прогнозирования.

Таким образом, Kaspersky MLAD состоит из различных модулей нейронных сетей, взаимодействующих друг с другом. Рассматривая модель прогнозирования данной работы под призмой Kaspersky MLAD, можно понять, что задача данной работы состоит в

том, чтобы создать один из этих модулей, выполняющий задачу прогнозирования. В дальнейшем планируется развитие данной работы и создание остальных модулей по примеру системы Kaspersky MLAD.

Но как именно система Kaspersky MLAD определяет аномалии. Для этого у неё есть пять видов детекторов аномалий. Но интересует нас лишь, так называемый, предиктивный детектор, который строится на основе нейронных сетей. Суть его работы проста: сначала выполняется прогноз данных, затем этот прогноз сравнивается с фактическим наблюдаемым поведением и с помощью машинного обучения определяются аномалии. Данный модуль можно разделить на две модели: модель прогнозирования и модель обработки прогнозов. Тем самым, данная работа, если рассматривать её под призмой системы Kaspersky MLAD, ставит своей целью создание первого модуля предиктивного детектора. Также нас интересует ещё один из видов детекторов аномалий, а именно потоковый процессор. Его задача состоит в том, чтобы приводить данные телеметрии к равноинтервальному временному виду, что подразумевает собой равным промежутки между точками относительно оси времени. Такой вид данных необходим для работы остальных видов детекторов аномалий, в том числе и для предиктивного детектора. Аномалии в потоковом процессоре вычисляются в процессе приведения данных к равноинтервальному временному виду.

Раз была определена схожесть данной работы с предиктивным детектором аномалий системы Kaspersky MLAD, тогда необходимо более подробно рассмотреть этот модуль и то, как он функционирует. Предиктивный детектор необходим для анализа данных технологического процесса, в основном представленных в виде временных рядов, содержащих разную информацию о состоянии объекта, различные физические величины, взятые с датчиков, и так далее. Предиктивный детектор работает по определённому алгоритму. Сначала формируется окно входных данных из поступивших на предиктивный детектор данных технологического процесса. На основе окна входных данных выполняется прогноз, формируя окно прогноза. Это окно предсказывает поведение объекта в определённом недалёком будущем. После вычисляется ошибка между прогнозом и реально наблюдаемыми данными для каждого предсказанного значения. Далее по совокупности этих ошибок производится вычисление среднеквадратичной ошибки. При вычислении этой ошибки значения индивидуальных ошибок прогноза умножаются на определённые веса. Таким образом осуществляется переход ко второй модели нейронной сети предиктивного детектора. Этот модуль определяется по значению среднеквадратичной ошибки аномалию с помощью некоторого порога, который формируется при обучении модели нейронной сети.

Предиктивный детектор позволяет снизить количество ошибочных оповещений об аномалиях технологического процесса, выявлять аномалии на ранних этапах развития проблемы с целью её предотвращения, а также обнаружить слабо проявляющиеся аномалии. Данная работа, хоть и не ставит своей целью именно поиск аномалий, однако известно, что её прогнозы будут в дальнейшем использоваться именно с этой целью. Этими словами мы однозначно закрываем первые этапы инженерии машинного обучения и построения прогноза.

Выполненные в ходе практики задачи

Преддипломная практика была пройдена в лаборатории БрГТУ «Системы идентификации и промышленная робототехника». В начале практики были поставлены различные задачи, связанные не только с самим дипломным проектом, но так же и с производством ОАО «Савушкин Продукт».

Основной задачей преддипломной практики являлась разработка комплекса программного обеспечения для прогнозирования данных временных рядов пастеризационной установки с целью предсказания будущего поведения системы и планирования производства. Данный комплекс ПО состоит из трёх модулей: модуля для анализа, обработки и подготовки данных к прогнозированию, модуля создания, обучения и тестирования моделей нейронных сетей, а также модуля, выполняющего непосредственное прогнозирование входящих в него данных.

Чтобы продолжить, необходимо разъяснить некоторые моменты, относящиеся к форматам данных. Для представления данных было разработано два формата: строчный формат OCDF и табличный формат TDF, данные которых находятся в аккуратном виде. OCDF-формат выглядит следующим образом:

<сид>;<время>;<значение>

В таком виде представлены начальные данные, которые были предоставлены ОАО «Савушкин Продукт». Данные формата TDF имеют следующий вид:

<время>;<зн1>;<зн2>;<зн3>;<зн4>;<зн5>;<зн6>

где, знН - значение данных под сидом Н.

Модуль для анализа, обработки и подготовки данных к прогнозированию предоставляет ряд функций для различного преобразования данных: обрезка данных по заданному количеству, заданному проценту и до или после указанного значения дискретного времени как справа, так и слева; визуализация данных для зрительного анализа и составления инфографики; добавление новых данных к уже имеющимся для создания необходимых датасетов для моделей нейронных сетей, использующих вспомогательные данные для построения прогноза; преобразование данных в равноинтервальный временной вид, что подразумевает выравнивание взаимного расстояния между соседними точками относительно оси абсцисс; парсинг данных по указанному сиду. Все эти функции применимы к данным как OCDF-формата, так и к данным TDF-формата, за исключением последней, поскольку данные TDF-формата уже находятся в равноинтервальном временном виде.

Также модуль даёт возможность создания TDF-формата из данных OCDF-формата. Для выполнения данной операции необходимо сделать большое количество шагов, задействующих все возможности модуля, которые перечислены выше.

Модуль создания, обучения и тестирования моделей нейронных сетей необходим для создания обученной модели нейронной сети для непосредственного прогнозирования данных как OCDF-формата, так и TDF-формата. Другими словами, благодаря этому модулю можно получить готовую к работе модель нейронной сети. Для создания такой модели предоставлены все возможности: непосредственное создание модели нейронной сети, в ходе которого имеется возможность указать её характеристики; обучение модели нейронной сети на наборах данных, подготовленных с помощью модуля для анализа, обработки и подготовки данных к прогнозированию; изменение характеристик модели нейронной сети; тестирование модели нейронной сети для настройки её гиперпараметров.

Модуль для непосредственного прогнозирования данных временных рядов пасте-

ризационной устновки говорит сам за себя о своём назначении. Этот модуль является целью данного дипломного проекта. Если два остальных модуля могут быть развёрнуты по сути где угодно, то модуль непосредственного прогнозирования как раз и должен быть развёрнут на программируемом микроконтроллере для выполнения основных задач по прогнозированию. Данный модуль должен быть частью системы, состоящей из двух или более моделей нейронной сети. Вторая часть этой системы разрабатывается моим коллегой Двораниновичем Дмитрием Александровичем.

Также в ходе преддипломной практики было начато глубокое погружение в изучение \LaTeX , в ходе которого удалось приобрести большой опыт использования данных средств. Отметим наиболее важные достижения в изучение \LaTeX :

1. Удалось разобраться, как подключать пользовательские шрифты. Для этого необходимо было не только скачать необходимые шрифты в формате .ttf, но также и получить метрику шрифтов в формате .tfm для возможности изменения размера символов шрифта, поскольку этот файл содержит информацию о ширине, высоте и глубине каждого символа; получить карту шрифта в формате .map для определения взаимного расположения символов; получить виртуальный шрифт в формате .vpl для его корректного отображения в pdf. Для того, чтобы получить всё вышеперечисленное необходимо было использовать консольную утилиту ttf2tfm.

2. Научился создавать собственные команды.

3. Создание рамок для пояснительно записки и титульного листа средствами, предоставляемыми движком \LaTeX . Для этого было использованы средства, позволяющие создавать графические объекты в \LaTeX .

4. Подробно изучил таблицы в \LaTeX для создания красивых разлиненных в нужном месте листов.

5. Понял, как использовать Inkscape для того, чтобы появилась возможность использовать векторные изображения в формате .svg.

Также были созданы средства автосборки проекта \LaTeX , а также документация к репозиторию на GitHub, где хранится исходный код дипломного проекта на \LaTeX .

Выполненные индивидуальной задачи

В ходе практики было уделено время и задачам ОАО «Савушкин Продукт», цель которых состоит в актуализации проекта SCADA-системы.

Прежде, чем перейти к описанию результатов выполненных задач, немного предыстории и некоторых пояснений. SCADA-система это огромный проект необходимый для контроля и управления производством. В этой системе можно создать модель любого технологического оборудования и эмулировать его работу, его технологический процесс. Данный проект берёт своё начало аж в 90-ые года, но и по сей день эта система используется на ОАО «Савушкин Продукт» и успешно справляется со своими задачами. Однако технологический процесс не стоит на месте: появляется новое компьютерное оснащение, операционные системы, новые версии IDE, стандарты и компоненты языка программирования и так далее. Задача актуализации проекта SCADA-системы заключается как раз именно в том, чтобы адаптировать эту систему под новейшие технологии и средства, исправив ошибки совместимости, а также заключается и в устранении сторонних компонентов, которые были использованы по тем или иным причинам при разработки системы и которые необходимо заменить на стандартные средства языка программирования.

Ранее мне была поставлена задача, которая заключается в удалении RX компонентов из SCADA-системы. Большая часть таких компонентов была заменена. В ходе практики было уделено время на устранение оставшихся RX-компонентов, а также написание тестов для заменённых на стандартные версии компонентов.

На данном этапе актуализации проекта SCADA-системы произошло знакомство с DUnit, они же unit-тесты для Delphi. В ходе создания проектов тестов пришлось столкнуться с рядом проблем, на решение которых ушло время, но это стало ценным опытом.

Во-первых, при создании проекта тестов требуется указать все те же связи, что есть и в проекте, для которого создаются тесты, а их могут быть десятки, что забирает некоторое время на настройку проекта теста.

Во-вторых, можно столкнуться с директивами препроцессора, которые перманентно могут вызывать ошибки компиляции проекта тестов. На первый взгляд, можно определить необходимые переменные, чтобы проходить или не проходить в определённые участки кода, но где это делать? А если таких переменных тоже десятки? Решение этой проблемы можно произвести относительно быстро: настроить Debugger для проекта тестов в соответствии с настройками Debugger тестируемого проекта. Это можно сделать следующим образом (пример на IDE Embarcadero Delphi XE5): Project → Options → Delphi Compiler → Conditional defines. Далее прописать там все определённые в тестируемом проекте переменные.

В-третьих, некоторые компоненты, например, TeeEngine, отказываются работать и выдают ошибку подключения своих библиотек, несмотря на то, что они числятся в компонентах проекта как подключённые или активные. Для решения данной проблемы необходимо настроить сам проект тестов в соответствии с тестируемым проектом. Это можно сделать следующим образом (пример на IDE Embarcadero Delphi XE5): Project → Options → Delphi Compiler → Unit scope names. Далее прописать там все определённые в тестируемом проекте зависимости. Если это не решило проекта, тогда необходимо прописать зависимости в Project → Options → Packages → Runtime Packages → Runtime Packages.

В-четвёртых, происходит автоматическая генерация кода с ошибками, которые также необходимо устранять. В основном ошибки генерации кода встречаются в файлах фор-

мата .dpr, т.е. в файлах проекта.

В-пятых, после всех настроек конфигурации проекта, всё равно будут различия между проектами, которые кроются в деталях. Одной из таких деталей является различие десятичного сепаратора. Его тоже необходимо настроить вручную, используя переменную `FormatSettings.DecimalSeparator`.

Преодолев все вышеописанные трудности, появилась возможность написания самих тестов. В ходе этого был разработан ряд тестов, проверяющих как заменённые визуальные компоненты, так и различные системные функции.

Для тестирования системных функций были созданы простейшие проекты средствами `delphi`. Это было сделано по причине того, что системные функции, которые необходимо проверить, являются частью других огромных функций или процедур, а протестировать необходимо лишь пару строчек.

Также мною была продолжена работа по замене сторонних компонентов на системные компоненты, которые предоставляет язык программирования `Delphi`.

Удаление `RX` компонентов из проекта `DBEditorXML` является непростой задачей. В этом проекте присутствует специфический компонент, аналогов которого стандартный `Delphi` предложить не может. Данный компонент использует `ini`-файлы и предназначен для сохранения и загрузки конфигурации окна.

Для замены этого компонента был проведён анализ `ini`-файла для того, чтобы заменить специфический компонент, средствами объекта `TIniFile`.

Были выявлены места, где производится автоматическая запись данных в `ini`-файл. Для этого один из сотрудников предоставил предзаполненную тестовую базу данных, которая открывается и редактируется средствами `DBEditorXML`. В выявленных местах были прописаны участки кода для работы с `ini`-файлом.

Также были выявлены места, где производится автоматическое считывание данных из `ini`-файла. В этих местах было прописано участки кода для считывания данных из `ini`-файла.