МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«ТЮМЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ НАУК
Кафедра программного обеспечения

РЕКОМЕНДОВАНО К ЗАЩИТЕ В ГЭК
Заведующий кафедрой
К. т. н., доцент
_____ М. С. Воробьева
23.06. 2023 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
магистерская диссертация

СОЗДАНИЕ И ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МОДЕЛИ ПРОГНОЗИРОВАНИЯ РЕЗУЛЬТАТОВ ПОВЕРКИ МЕТРОЛОГИЧЕСКОГО ОБОРУДОВАНИЯ

02.04.03 Математическое обеспечение и администрирование информационных систем

Магистерская программа «Разработка технологий Интернета вещей и больших данных»

Выполнил работу
студент 2 курса
очной формы обучения
Левкин Дмитрий Андреевич

Научный руководитель
доцент кафедры программного обеспечения, к. ф.-м. н.,
Ступников Андрей Анатольевич

Рецензент
доцент, к. т. н., зав. кафедрой информационной безопасности
Оленников Алексей Александрович

Тюмень
2023

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

В России существует процесс поверки, представляющий собой совокупность операций по подтверждению соответствия метрологических характеристик рабочих средств измерений (СИ) путем сравнения фактических значений со значениями эталона [1]. Суть поверки в следующем:

1) СИ признается годным к работе в течение определенного межповерочного интервала, если поверка подтвердила соответствие метрологическим характеристикам, указанным в паспорте СИ [2].

2) Если средство измерений признано непригодным, то оно направляется в ремонт, для восстановления необходимых метрологических характеристик, а затем снова проходит процедуру поверки, но уже более серьезной (первичной поверки). В случае, если СИ не подлежит ремонту, то заменяется новым.

Данный процесс схематично представлен на рисунке 1.

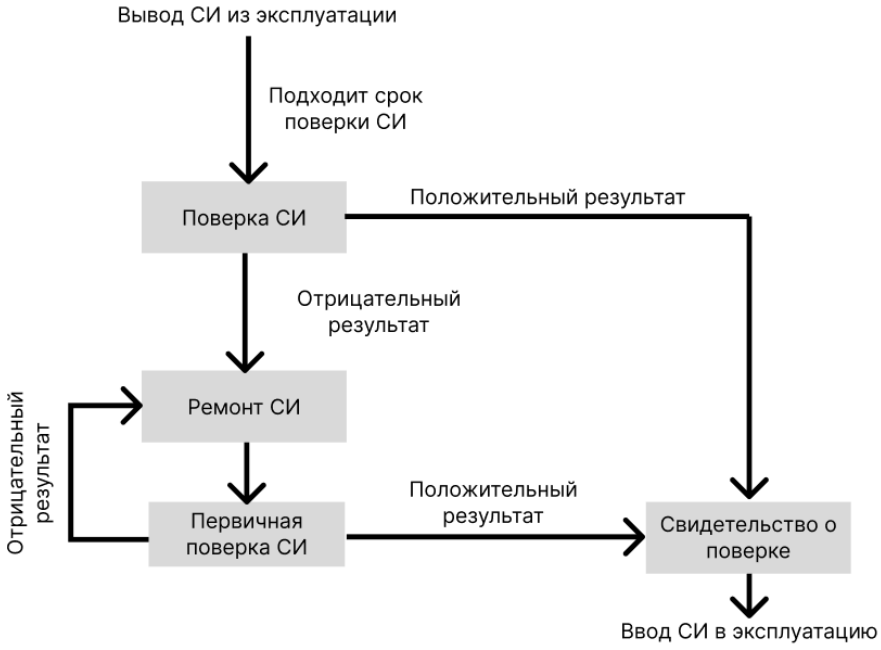

Рис. 1. Процесс поверки

Практически все компании, которые используют различные счетчики, манометры, датчики и т.д. должны соблюдать 184 федеральный закон о техническом регулировании. Он устанавливает обеспечение защиты здоровья и

жизни населения, флоры и фауны, окружающей среды, как основную цель. Охрана осуществляется от разного рода действий, которые могут нанести ущерб или ввести в заблуждение. Главными инструментами являются стандартизация и метрология. [3]

Поэтому все средства измерения, которые используются в производственном процессе косвенно или непосредственно, обязаны иметь свидетельство о поверке. Данный процесс крайне затратный с точки зрения бизнеса, так как на время поверки необходимо останавливать процесс, в котором состоит данное средство измерение. Крупные компании, конечно, так не делают – они заменяют тот или иной прибор на «запасной» на время проведения поверки, таким образом практически не прерывая производственный процесс. Однако в таком случае компании нужно проводить поверку (а иногда и ремонт) двух приборов на одно и то же место, что соответственно увеличивает стоимость в 2 раза.

Актуальность выбранной темы обосновывается необходимостью прогноза будущих результатов поверки (определение параметров, которые на него влияют) для составления экономического плана предприятия по обеспечению единства измерений, потому что процедура поверки/ремонта или замены СИ в промышленных масштабах имеет большую финансовую нагрузку для предприятий, которые используют СИ в производственном процессе.

Научная новизна исследования состоит в применении методов машинного обучения для получения вероятностной оценки применительно к результатам будущей поверки технологических приборов учета.

Основной целью работы было исследование эффективности математических методов, которые позволяют получить вероятностную оценку результатов поверки. Практическая значимость состояла в том, что найденный метод будет использоваться в модуле поддержки принятия решения для комплексной оценки вероятности результатов поверки СИ, которая предназначена для работников участвующих в составлении годового плана закупок (ГПЗ) предприятия.

Работа выполнялась на данных, которые были получены на предприятии ООО «Газпромнефть-Автоматизация».

Исследование данной темы позволит рациональнее распределить временные и материальные ресурсы предприятия-владельца СИ, что в свою очередь будет способствовать повышению прибыли и/или экономии бюджетных средств. Поскольку государство регулирует большинство сфер жизни, конечный продукт данного исследования будет полезен во многих областях, например, в таких, как нефтегазовая, фармацевтическая, пищевая и др.

Для достижения поставленной цели были обозначены следующие задачи:

1) Работа с бизнес требованиями:
    a. Определение сути проблемы, ответ на вопросы;
        i. Что мы хотим получить?
        ii. Для чего нам это нужно?
    b. Определение средств и инструментов;
2) Работа с данными:
    a. Описание и подготовка имеющихся данных;
        i. Объединить значения отдельных признаков схожих по смыслу, но различных в написании;
        ii. Посчитать текущие отклонения измерений;
        iii. Определить выбросы;
        iv. Оценить корреляцию признаков друг с другом;
        v. Сконструировать новые признаки.
    b. Определение необходимости использования сложных моделей;
3) Разработка:
    a. Создание моделей машинного обучения;
        i. Оценить точность;
        ii. Сформулировать выводы о результатах;
        iii. Сравнить и выбрать модель, которая удовлетворяет поставленным требованиям.
    b. Объяснение поведения выбранной модели.

Суть исследования – это построение математической модели, которая позволяет оценить вероятность результатов следующей поверки СИ. Метод математического моделирования предполагает создание наглядно-образной характеристики, например, с помощью схем и математических формул. Эффективным инструментом для такого моделирования стали методы машинного обучения. Одним из которых является регрессия.

Определить следующий результат поверки можно с помощью регрессионной модели, которая описывает зависимость параметров СИ от погрешности измерений. Цель регрессии — это подобрать такую математическую функцию, которая будет описывать связь всех переменных, характеризующих различные состояния одного объекта по формуле:

$$y = f(x) + \varepsilon,$$

где y – целевая переменная,

x – вектор признаков,

ε – случайная величина.

Задача регрессии на практике – это прогнозирование значения целевой переменной по известному набору параметров. Пример регрессионной зависимости представлен на рисунке 2.

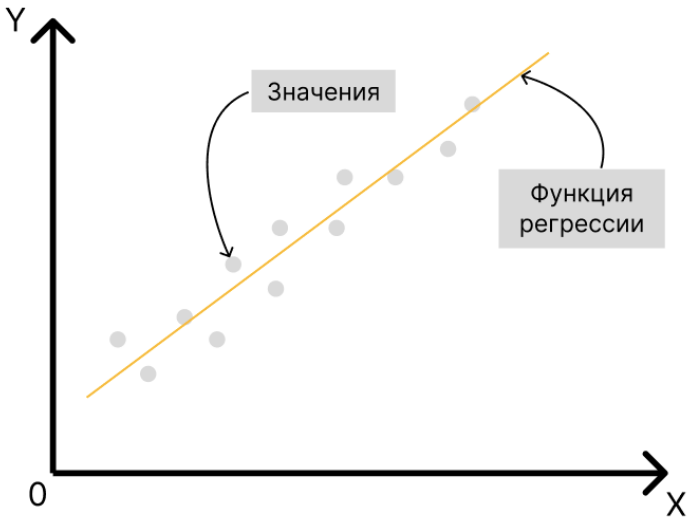

Рис. 2. Пример регрессионной зависимости целевой переменной Y от признака X

Сейчас машинное обучение активно применяется на практике в различных сферах, в том числе и на производстве. Но в большинстве случаев его применение ограничивается задачами классификации. Задача классификации – это присвоение меток (классов) объектам классификации на основе их сходства с известными классами объектов, но в отличии от задачи регрессии в классификации мы получаем ответ с определенной вероятностью.

Классическая модель регрессии выдает нам только предсказанное значение целевой переменной, а вот модель классификации выдает метку класса и вероятность, с которой данный объект относится к выбранному классу.

С точки зрения бизнеса, нам было необходимо создать основу инструмента, который позволил бы получить вероятность, с которой значение погрешности следующего результата измерения войдет в диапазон допустимых значений (вероятность необходима, потому что, в конечном итоге, решение будет принимать человек).

Предположительно, для подобной реализации понадобятся следующие технологии:

Для реализации моделей машинного обучения:

1) Язык программирования высокого уровня Python версии 3.10;
2) Открытая программная библиотека глубокого обучения TensorFlow.

Для создания пользовательского интерфейса:

1) Объектно-ориентированный язык JavaScript;
2) React – JavaScript-библиотека для создания интерфейсов.

ГЛАВА 1. РАБОТА С ДАННЫМИ

1.1. ОПИСАНИЕ ИСХОДНЫХ ДАННЫХ

На основании результатов предыдущих исследований было установлено, что создание единой модели для всех категорий СИ практически невозможно, так как СИ из разных функциональных групп могут полностью отличаться друг от друга. Мы приняли решение, что в рамках данной работы ограничимся одной группой – СИ давления. Сюда относятся различные датчики давления, манометры и вакуумметры.

Первоначальный набор данных был получен в ООО «Газпромнефть-Автоматизация» в виде файла табличного формата .xlsx. Данный набор содержит информацию о 10602 образцах СИ. Данные были выгрузкой из корпоративной системы учета, поэтому там уже нет пропущенных значений ввиду функциональной специфики системы.

Полученные данные представлены в виде таблицы со следующими столбцами:

1) Месторождение (field);
2) Площадка на месторождении (prod_area);
3) Тип СИ (type);
4) Дата выпуска СИ (release_date);
5) Дата последней поверки СИ (last_ver_date);
6) Организация, в которой проводилась последняя поверка СИ (organization);
7) Номер типа СИ в государственном реестре утвержденных типов СИ (num_registry);
8) Результаты контрольных измерений (measurement_1 – measurement_20);
9) Результат прошлой поверки (pre_result) в формате: 1 – означает, что СИ прошло прошлую поверку с первого раза, 0 – СИ было в ремонте во время прошлой поверки.

1.2. РАСШИРЕНИЕ НАБОРА ДАННЫХ

Для подробного описания образцов, набор данных был расширен дополнительной информацией. Это нужно для того, чтобы модель «понимала реальную картину мира». Дополнительная информация собрана из открытого Федерального информационного фонда по обеспечению единства измерений с официального электронного ресурса, который доступен по ссылке: **https://fgis.gost.ru/fundmetrology/registry**. Данная информация добавлялась к существующей на основе номера типа СИ в государственном реестре утвержденных типов СИ как «первичного ключа». Были добавлены следующие столбцы-признаки средств измерений:

1) Страна-производитель СИ (country);
2) Завод-изготовитель СИ (factory);
3) Срок службы СИ (life_time);
4) Межповерочный интервал (interval);
5) Верхний и нижний пределы измерений (upper_limit, lower_limit);
6) Единицы измерений (units);
7) Относительная погрешность СИ (relative_error).

Данные подверглись «очистке» средствами MS Excel и Python, ее этапы представлены в таблице 1.

Таблица 1

Этапы «очистки» данных

Этап	**Что было сделано**
Удаление лишних пробелов	Были удалены пробелы вначале и конце слов и предложений. Удалены двойные пробелы между словами.
Выявление отдельных сущностей	Для объединения значений по смыслу из наименования исключались некоторые слова, например, дублирование производителя. Или заменялись слова, например, «преобразователи давления» и «датчики давления» объединялись в «датчики давления». Из названий производителя исключались различные сокращения (ООО, АО, ЗАО и др.)

Продолжение таблицы 1

Этап	Что было сделано
Форматирование дат	Все даты, которые встречались в наборе были приведены к шаблону ДД.ММ.ГГГГ
Приведение к одним единицам измерений	Так как у нас были результаты измерений, нам необходимо было привести их к единой шкале. Для этого мы добавили к данным единицы измерения на основе государственного реестра, а затем перевели значения результатов измерений, верхнего и нижнего предела измерений в бары.

Описание признаков набора исходных данных представлено в таблице 2.

Таблица 2

Описание исходных данных

Признак	Описание	Тип
field	Номер месторождения (настоящие названия заменены в целях конфиденциальности) Всего 6 уникальных значений, наиболее часто встречаемое – месторождение №5	string
prod_area	Названия площадок (настоящие названия заменены в целях конфиденциальности) Всего 43 уникальных значения, наиболее часто встречаемое – площадка «Куст скважин»	string
type	Тип СИ Всего 36 уникальных значений, наиболее часто встречаемое – «Метран»	string
country	Страна-изготовитель Всего 8 уникальных значений, наиболее часто встречаемое – «Россия»	string
factory	Завод-изготовитель Всего 35 уникальных значений, наиболее часто встречаемое – «Метран»	string

Продолжение таблицы 2

Признак	**Описание**	**Тип**
life_time	Срок службы, в годах Наиболее часто встречаемое – «12» лет, минимальное значение - «1» год, максимальное – «50» лет	float
interval	Межповерочный интервал, в месяцах Наиболее часто встречаемое – «60» месяцев, минимальное значение - «12» месяцев, максимальное – «72» месяца	int
organization	Организация, которая проводила последнюю поверку Всего 31 уникальное значение, наиболее часто встречаемое – «Тюменский ЦСМ»	string
lower_limit	Нижний предел измерений, бар. Наиболее часто встречаемое – «0», минимальное значение – «-5000», максимальное – «1000»	float
upper_limit	Верхний предел измерений, бар. Наиболее часто встречаемое – «250», минимальное значение – «0,001», максимальное – «20274.509804»	float
relative_error	Относительная погрешность, в процентах Наиболее часто встречаемое – «25», минимальное значение – «0,075», максимальное – «3»	float
num_registry	Номер типа СИ в государственном реестре утвержденных типов СИ Всего 105 уникальных значений, наиболее часто встречаемое – «32854-13»	string
pre_result	Предыдущий результат поверки Всего 2 уникальных значения, наиболее часто встречаемое – «0» «0» - СИ прошло поверку без происшествий, «1» - СИ не прошло поверку с первого раза	int
release_date	Дата выпуска СИ Диапазон – с «01.01.1984» по «10.12.2021»	date
last_ver_date	Последняя дата поверки СИ Диапазон – с «27.02.2015» по «04.12.2022»	date

1.3. ФОРМИРОВАНИЕ НОВЫХ ПРИЗНАКОВ

Следующим этапом было конструирование и замена признаков на модифицированные. Конструирование признаков является одной из важнейших задач машинного обучения, потому что используемая модель располагает только той информацией, что мы ей предоставим – ее картина мира ограничена задачей. Классификация признаков представляет собой шкалу определений от «строго релевантные» до «нерелевантные». «Строго релевантные» – признаки, которые содержат информацию не доступную в других признаках. «Нерелевантные» – признаки, содержащие информацию, которая может быть в других признаках [4].

Как правило, для лучшего результата обучения модели нужно чтобы признаки были независимы между собой (но имели связь с целевой переменной). На основе описанной шкалы и данного правила мы сконструировали новые признаки и убрали те, которые учли в новых. Добавлены следующие признаки:

1) Следующая дата поверки (next_ver_date):

К дате последней поверки был прибавлен межповерочный интервал. Диапазон значений полученного признака – с «27.02.2015» по «04.12.2022».

2) Относительная погрешность измерений, числовой признак:

По формулам (1.1) и (1.2) была посчитана относительная погрешность каждого измерения (всего получилось 20 значений – по одному для каждого измерения):

$$KT_i = (up_i - low_i) * 0.7 + low_i, \quad (1.1)$$

где *low* – нижний предел диапазона измерений i-го СИ,

up – верхний предел диапазона измерений i-го СИ,

i – номер СИ.

$$\delta_{i,j} = \frac{measurement_{i,j} - KT_i}{KT_i}, \quad (1.2)$$

где КТ – эталонное контрольное значение i-го СИ,

measurement – значение j-го измерения i-го СИ.

3) Размах измерений, числовой признак (amplitude):

Данный признак получился в результате вычитания минимального значения ряда из максимального. Диапазон значений признака – от «0.000003» до «36.938826».

4) Стандартное отклонение ряда измерений, числовой признак (std) (1.3);

$$\sigma_i = \sqrt{\frac{\sum(x_{i,j} - \overline{x})}{n-1}}, \quad (1.3)$$

где x – значение относительной погрешности j-го измерения,

$\bar{x}$ – среднее значение измерений,

i – номер СИ,

j – номер измерения.

5) Коэффициент точности, числовой признак (accuracy_coef):

Данный коэффициент показывает зависимость максимальной допустимой относительной погрешности от диапазона измерений и рассчитывается по формуле (1.4).

$$acc_i = \left|\frac{\delta max_i}{(up_i - low_i)}\right|, \quad (1.4)$$

где low – нижний предел диапазона измерений i-го СИ,

up – верхний предел диапазона измерений i-го СИ,

δ_{max} – максимальная допустимая относительная погрешность i-го СИ.

Модифицированы следующие признаки:

1) Единицы измерений (units):

Так как на этапе очистки данных мы преобразовали значения измерений и пределов к одной шкале, то единицы измерений у нас теперь одни. Соответственно, в данном признаке больше нет необходимости.

2) Признаки измерений «measurement_1- measurement_2»:

Данные признаки уже учтены в относительной ошибке и статистических признаках ряда, так что их можно отбросить.

3) Столбцы с типом данных «date»:

Все даты мы перевели в UNIX секунды.

1.4. ОПРЕДЕЛЕНИЕ КОРРЕЛЯЦИИ ПРИЗНАКОВ

Корреляция – это инструмент, который помогает выявить связи между двумя случайными величинами. Для эффективного обучения моделей машинного обучения на вход им необходимо подавать признаки, которые не коррелируют между собой, так как в ином случае такие признаки будут давать модели, практически, одну и ту же информацию. В нашем случае, для отбора признаков мы использовали коэффициенты корреляции.

Признаки, описывающие «неметрологические» характеристики СИ относятся к категориальным. В наборе данных они кодируются числами, но эти множества «бессмыслены», так как в них нет понятий «больше» или «меньше».

Для определения связи между категориальными признаками строятся таблицы сопряженности, где на пересечении граф с признаками стоит V-коэффициент корреляции Крамера (мера тесноты связи, используется для таблиц, где есть более 2-х признаков). Он основан на критерии хи-квадрата Пирсона и был опубликован Харальдом Крамером в 1946 году и вычисляется по формуле (1.5) [5]:

$$V(x,y) = \sqrt{\frac{X^2/n}{\min(r-1;c-1)}}, \quad (1.5)$$

где X^2 – критерий хи-квадрат Пирсона, рассчитывается по формуле (1.6),

n – общий размер выборки,

r – количество рядов,

c – количество столбцов,

x, y – случайные величины, корреляционная связь которых определяется.

$$X^2(x,y) = \sum_{i,j} \frac{(n_{ij} - {n_I n_J}/{n})^2}{{n_I n_J}/{n}}, \quad (1.6)$$

где i – для x, диапазон значений 1,…,r,

j – для y, диапазон значений 1,…,c,

n_{ij} – количество раз, когда наблюдались значения $(x_i; y_j)$,

$n_I = \sum_j n_{ij}$ – количество наблюдений за значением x_i,

$n_J = \sum_i n_{ij}$ – количество наблюдений за значением y_i.

Данный коэффициент принимает значения в диапазоне [0, 1], где 0 означает, что признаки статически независимы, 1 – характеризует наличие связи (функциональную).

Для анализа зависимости признаков мы реализовали на Python построение тепловой карты V-коэффициентов Крамера. Тепловая карта (heatmap) – это способ визуализации данных, который представляет собой таблицу величин в виде цвета.

Полученная тепловая карта представлена на рисунке 3.

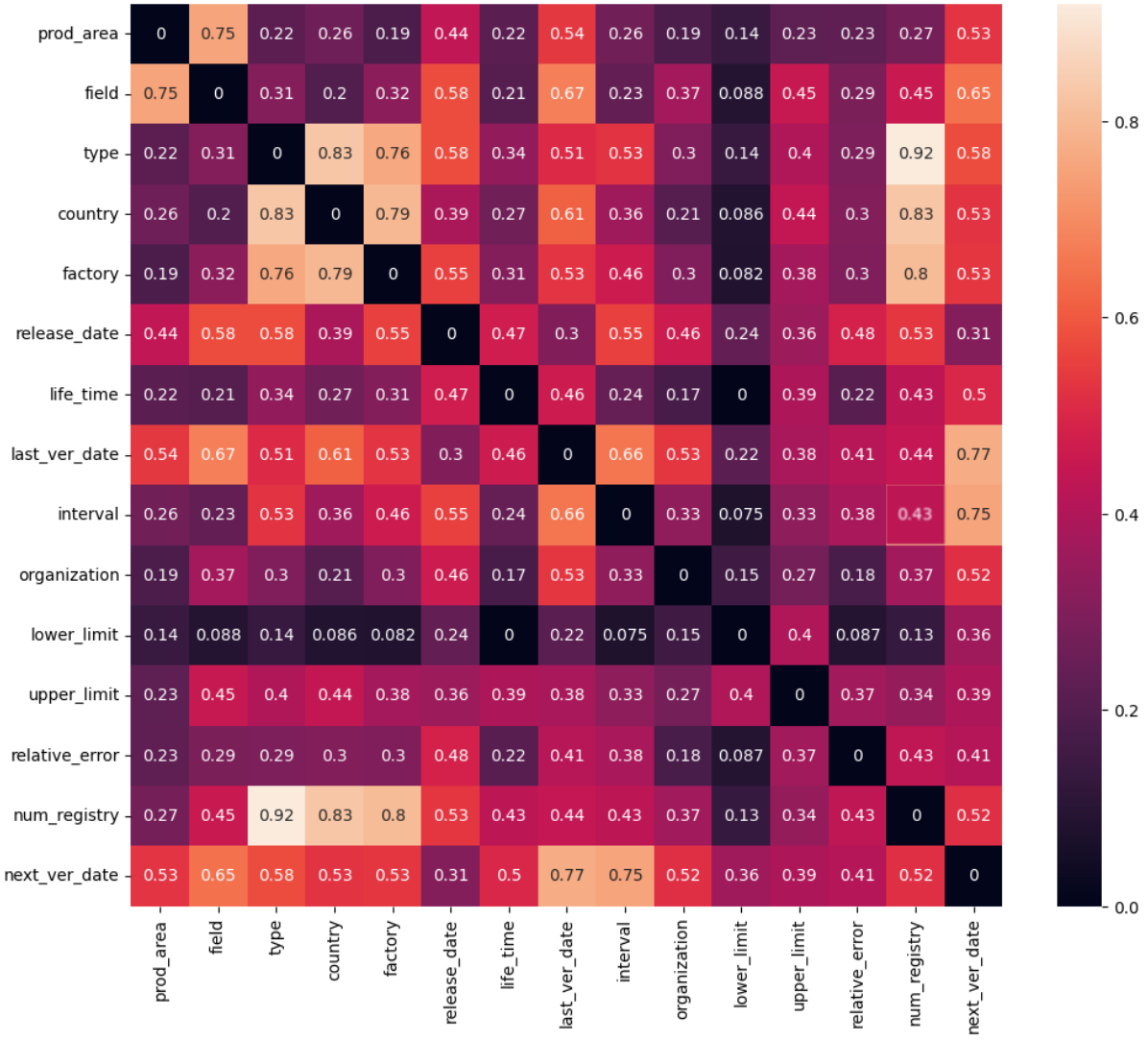

Рис. 3. Тепловая карта коэффициентов корреляции Крамера

По данному рисунку мы видим, что заметная корреляционная связь присутствует только у признаков с типом «date» и категориальных признаков, которые зависят от номера государственного реестра. Соответственно, признаки «Тип СИ», «Страна-производитель» и «Завод-изготовитель СИ» уже «включены» в суть государственного реестра и их можно исключить. Признаки «Следующая дата поверки» и «Последняя дата поверки» коррелируют, так как они связаны через «Межповерочный интервал» – СИ подвергаются первичной поверке при производстве, соответственно, с этой даты прибавляется межповерочный интервал.

ГЛАВА 2. ОБОСНОВАНИЕ ВЫБОРА МЕТОДОВ ПРОГНОЗИРОВАНИЯ

2.1. ОПРЕДЕЛЕНИЯ ЗАКОНА РАСПРЕДЕЛЕНИЯ ДАННЫХ

В связи с тем, что применение средств математической статистики требует от обрабатываемых данных соответствия нормальному закону распределения, то в качестве первой подзадачи была выбрана проверка данных о погрешности СИ на нормальность. Для этого существуют два основных способа, мы использовали оба:

1) Графический (построение гистограммы и график квантиль-квантиль (QQ);

2) С помощью статистических тестов (мы взяли критерий Шапиро-Уилка) [6].

Чтобы применить статистические критерии необходимо было выдвинуть две гипотезы и установить уровень значимости:

1) Нулевая гипотеза: данные подчиняется нормальному закону распределения;

2) Альтернативная гипотеза: данные не подчиняется нормальному распределению;

3) Установили уровень значимости равным 0.1.

Нормальное распределение – это распределение, которое задается функцией Гаусса. В идеальном мире такое распределение симметрично и является функцией плотности само для себя. Это означает, что вероятность попасть в интервал (x, y) равна площади под кривой нормального распределения в интервале (x, y). На рисунке 4 представлена гистограмма распределения значений погрешности измерений.

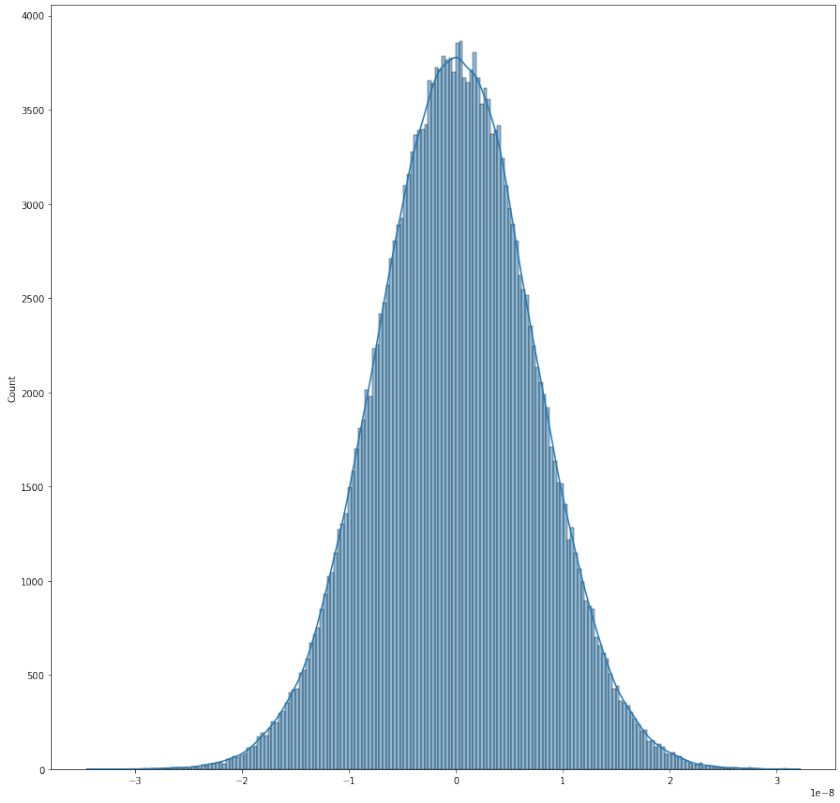

Рис. 4. Гистограмма распределения погрешности измерений

Гистограмма достаточно хорошо напоминает нормальное распределение, за исключением небольшой кривизны сверху. Слепо верить только этому методу нельзя, поэтому так же мы построили QQ-график. QQ-график – это графический метод сравнения двух распределений путем построения их квантилей, он позволяет оценить отклонения реальных данных от теоретического распределения. Данный график представлен на рисунке 5.

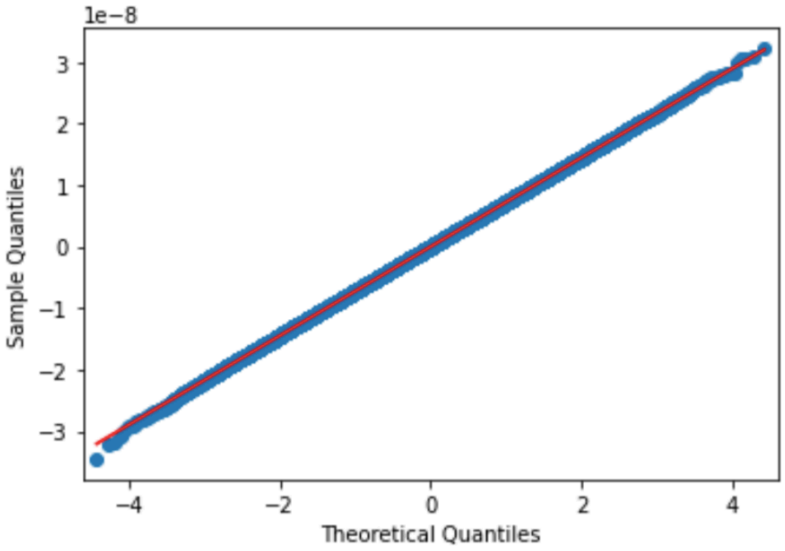

Рис. 5. QQ-график распределения вероятности

Данный график сравнивает два распределения, теоретическое и эмпирическое. В данном случае это наше реальное распределение с моделируемым распределение на основе мат. ожидания и стандартного отклонения нашего ряда [7]. Когда два распределения подобны, то точки на графике лежат на прямой. За исключением концов, наше распределение близко к нормальному.

Чтобы окончательно принять или опровергнуть нулевую гипотезу мы воспользовались критерием Шапиро-Уилка. Этот критерий является отношением линейной оценки дисперсии к ее оценке методом максимального правдоподобия. Он рассчитывается по формуле (2.1):

$$W = \frac{1}{s^2}\sum_{i=0}^{n} a_{n-i+1}(x_{n-i+1} - x_i), \quad (2.1)$$

где a – уровень значимости,

n – размер выборки,

s^2 – среднеквадратичная оценка Ллойда, которая рассчитывается по формуле (2.2)

$$s^2 = \sum_{i=0}^{n}(x_i - \bar{x}^2), \quad (2.2)$$

где x – входное значение.

Если критерий дает оценку больше принятого уровня значимости, то принимается нулевая гипотеза, если меньше – нулевая гипотеза отвергается. В результате расчетов мы получили значение равное 0,76. Это сильно больше приятого уровня значимости, значит мы можем принять нулевую гипотезу – данные о погрешности измерений распределены нормально.

Таким образом, мы считаем, что все значения погрешности наших образцов распределены нормально. К тому же, распределение значений измерений с ошибками (погрешностью) естественным образом стремится к нормальному.

2.2. СПОСОБ ОПРЕДЕЛЕНИЯ ВЕРОЯТНОСТИ

На этом этапе был сформулирован способ определения вероятностной оценки результатов регрессионной модели. Главная задача — это не просто выдать число, а определить вероятность, с которой в действительности это число попадет в допустимый интервал. На рисунке 6 наглядно проиллюстрирована данная задача.

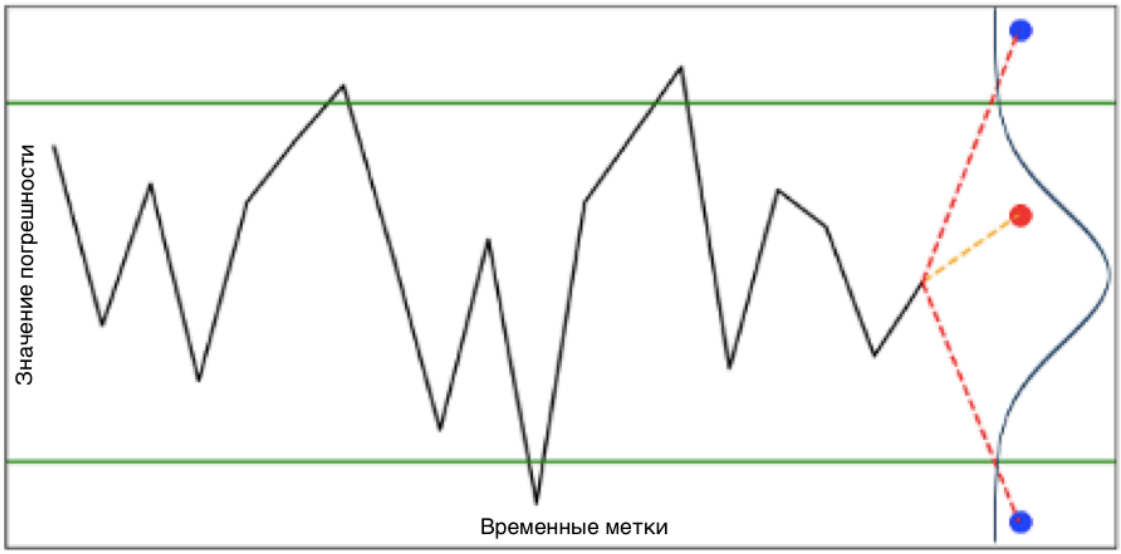

Рис. 6. Схематичное представление задачи

Черная линия отражает динамику значений погрешности измерений некоторого отдельного прибора, точки – это прогнозируемые значения, зеленые рамки – это допустимый диапазон значений, синяя линия – это моделируемое распределение значений.

Так как прогнозируемое значение X – случайная непрерывная величина, то мы не можем оценить вероятность того, что X будет равна истинному значению X’ с какой-либо вероятностью. Точнее эта вероятность будет равна нулю, но здесь имеет смысл другая форма – принадлежность X к интервалу $[X'–\delta; X'+\delta]$, где δ некоторое малое значение. Такое событие может иметь ненулевую вероятность [8].

Существует два основных подхода, которые могут помочь в определении данной вероятности:

1) Метод сравнение с прошлыми образцами;

2) Метод моделирования плотности распределения.

Первый способ крайне затратный в плане набора данных, для него необходимо было бы иметь для всех оцениваемых образцов большое количество значений, которые предшествовали данной временной точке. Поэтому мы приняли решение использовать второй подход. Так мы сможем учитывать описательные параметры каждого прибора.

Наиболее доступный способ получить плотность распределения вероятностей – это генерация ее параметров [9]. Функция распределения должна быть монотонной и её предельные значения на минус бесконечности и на бесконечности должны быть 0 и 1. Плотность нормального распределения задается функцией $y = f(x|\mu, \sigma)$, где x – представляет все возможные значения целевой переменной y, а μ и σ описывают точную форму распределения. Функция плотности нормального распределения имеет вид [10] (2.3):

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad (2.3)$$

где σ – стандартное отклонение,

μ – мат. ожидание.

По заданному набору мы решили «прогнозировать» не конкретное значение, а параметры функции плотности распределения – стандартное отклонение и мат. ожидание.

Для этого мы использовали нейронные сети. Понятие нейронной сети в машинном обучении появилось в 1940-х после того, как произошел бум популярности исследований на понимание работы мозга. Нейронная сеть в машинном обучении – это математическая модель принципа работы настоящей нейронной сети мозга. Нейронная сеть состоит из юнитов, их еще называют нейронами.

Схематично нейрон представлен на рисунке 7.

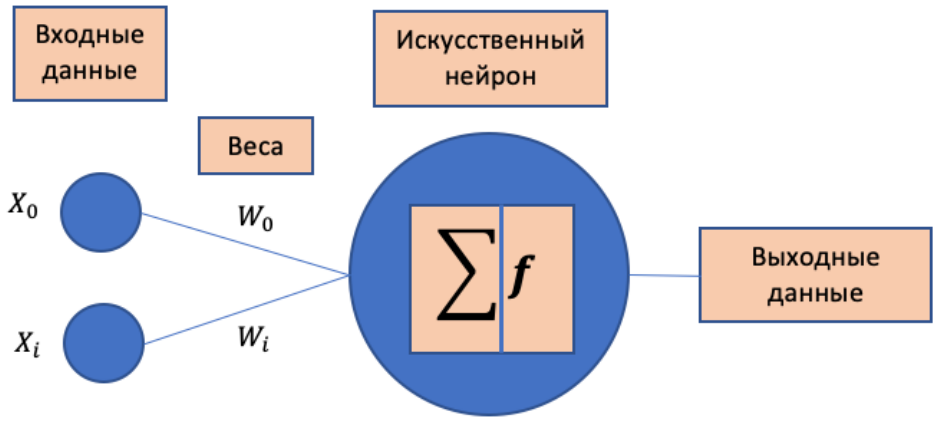

Рис. 7. Схематичное представление искусственного нейрона

Внутри реального нейрона величина собранного со всех входов электрического сигнала сравнивается с некоторым пороговым значением, индивидуальным для каждого нейрона, после преодоления которого включается аксон и передает сигнал другим нейронам. В искусственных нейронах этот процесс эмулируется с помощью функции активации. Функция активации преобразует значения, полученные из предыдущего нейрона, то есть является зависимостью формы сигнала на выходе от сигналов на входе нейрона, форма преобразования зависит от задачи.

Чтобы удовлетворить условия прогнозируемых значений был создан дополнительный специальный выходной слой нейронной сети, который использует две функции активации:

1) linear (2.4) – для мат. ожидания. Выходное значение равно входному, Она вычисляется по формуле (2.4).

$$y(x) = x, \tag{2.4}$$

где x – входное значение.

2) ELU+1 – для параметра «*sigma*». +1, потому что мы прибавляем к значению функции 1, так как нам необходимо неотрицательное значение, полученная формула функции активации представлена формулой (2.5) [11]:

$$\mathrm{y}(x) = \begin{cases} x + 1, x \geq 0 \\ e^x, x < 0 \end{cases}, \tag{2.5}$$

где x – взвешенная сумма входных значений.

Графическое представление полученной функции активации изображено на рисунке 8.

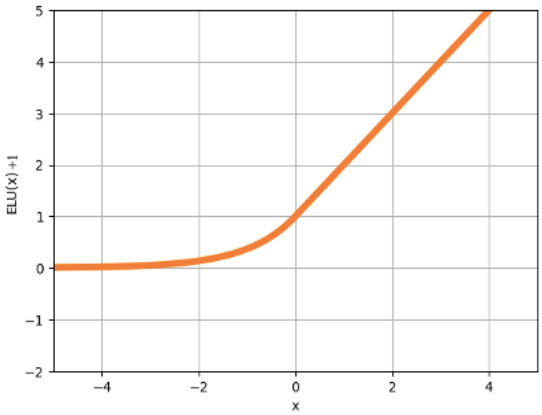

Рис. 8. Функция активации слоя для параметра «sigma»

В процессе обучения нейронная сеть подбирает веса W_i с целью минимизации значений функции потерь. Функция потерь используется для расчета «ошибки» между реальными данными и предсказанными ответами. Простыми словами, функция потерь оценивает, насколько хорошо работает нейронная сеть.

В качестве функции потерь использовалась реализация «оценки максимального правдоподобия». Смысл в том, что внутри этой функции «формируется» распределение с полученными параметрами (дисперсия и математическое ожидание), затем вычисляется вероятность того, что наша целевая переменная принадлежит к полученному распределению, то есть нам нужно получать как можно большую вероятность.

В процессе обучения нам необходимо минимизировать значение функции потерь с помощью определенного алгоритма, который называется оптимизатором нейронной сети. Оптимизатор нейронной сети – это своеобразная «функция активации» для целой сети, по определенной формуле обновляются параметры (например, веса и скорость обучения) нейронной сети по алгоритму градиентного спуска. [12] Градиентный спуск – это способ нахождения локального минимума функции, а сам градиент – это вектор, который указывает направление спуска относительно какой-либо точки. Иными словами, градиентный спуск – это метод последовательного изменения набора

(вектора) весов входных сигналов, где каждый следующий набор выбирается в направлении минимального значения функции потерь.

Так как оптимизатор нейронной сети направлен на минимизацию функции потерь, то мы берем значение со знаком «минус». Так как речь идет о малых вероятностях, то нам следует прологарифмировать полученное значение, для простоты интерпретации результата функции потерь. Полученная функция представлена формулой (2.6) [13]:

$$f(X') = -log\left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(X\prime-\mu)^2}{2\sigma^2}}\right), \quad (2.6)$$

где σ – стандартное отклонение,

μ – мат. ожидание,

X’ – реальное значение.

После получения параметров функции распределения относительной погрешности измерений мы рассчитываем вероятность попадания значения следующего измерения в допустимый интервал. Для получения этого значения мы интегрировали функцию плотности нормального распределения в искомом интервале (2.7).

$$p(x) = \int_a^b \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad (2.7)$$

где σ – стандартное отклонение,

μ – мат. ожидание,

a – нижняя граница интервала,

b – верхняя граница интервала.

В нашем случае вероятность попадания в интервал равна площади под кривой данной функции.

2.3. ИСПОЛЬЗУЕМЫЕ АРХИТЕКТУРЫ НЕЙРОННОЙ СЕТИ

Архитектура нейронной сети – это графическое или текстовое описание последовательного и/или параллельного перехода входных данных между различными функциональными группами нейронов (слоев).

За основу брались две архитектуры нейронных сетей:

1) С полносвязным скрытым слоем (Dense);

Это слой, нейроны которого связаны со всеми нейронами предыдущего слоя. На данном слое выполняется матричное перемножение элементов, которые пришли с предыдущего слоя со своими весами, математически это можно представить формулой (2.8).

$$y = f(W * x + b), \tag{2.8}$$

где f – функция активации,

W – матрица весов,

b – смещение,

x – сумма входных значений,

y – выходное значение.

Смещение в данной формуле – это определенная константа, которая добавляется к каждому слою нейронной сети, чтобы значения функции активации свободно перемещались по оси X.

2) Со скрытым слоем, который использует ячейки долгой краткосрочной памяти (LSTM).

Этот слой является разновидностью рекуррентных нейронных сетей. Смысл рекуррентных сетей (RNN) – возвращение значения слоя обратно на этот же слой для использования при обработке следующего пакета входных сигналов слоя. Проблема RNN-сетей в том, что они не обладают «долгой» памятью, то есть по мере удаления актуальной информации от момента использования информации сети теряют способность связывать информацию. Основная идея LSTM-сетей (сети долгой краткосрочной памяти) – это, как понятно из названия, долгосрочное запоминание зависимостей. LSTM-архитектуру представили Хохрайтер З. и Шмидхубер Ю. в 1997 году и схематично представлена на рисунке 9 [14].

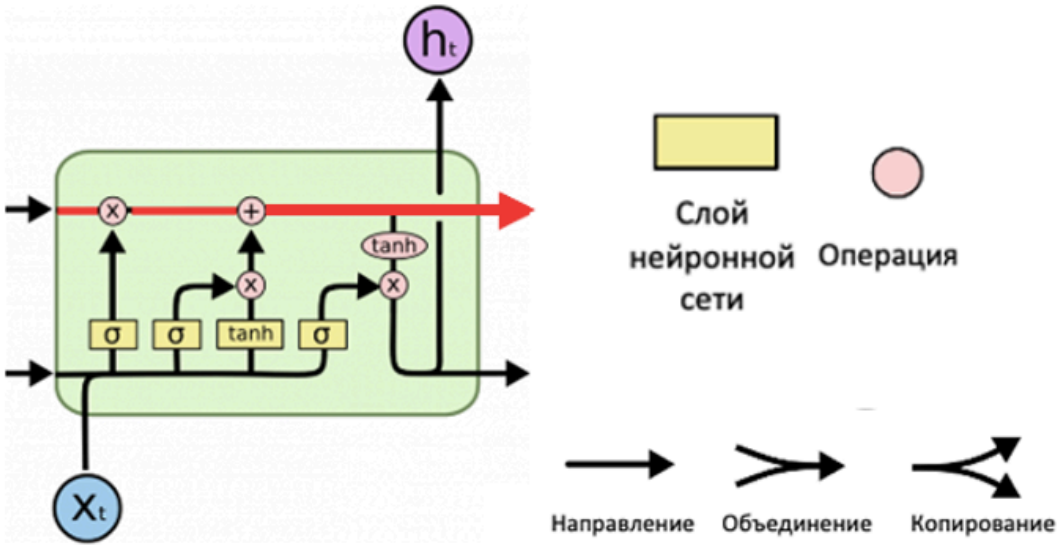

Рис. 9. Схематичное представление LSTM-архитектуры

Долгосрочная память достигается путем наличия состояния ячейки – красная линия на рисунке выше, по ней данные проходят без изменений.

2.4. ЛОГИСТИЧЕСКАЯ РЕГРЕССИЯ

Чтобы обосновать необходимость использования именно нейронных сетей в данной задаче, мы также использовали простой алгоритм машинного обучения, который называется «логистическая регрессия». Логистическая регрессия – это статистический метод классификации с использованием линейного дискриминанта Фишера [15]. Основная идея логистической регрессии состоит в том, что значение прогнозируемой функции проходит через функцию сигмоиды, по формуле (2.9), которая возвращает число от 0 до 1 и характеризует вероятность того, что объект относится к классу 1.

$$sigm(x) = \frac{1}{1+e^{-x}}, \tag{2.9}$$

где x – входное значение нейрона.

В готовом наборе данных у нас был признак *pre_result* (результат прошлой поверки), который принимает одно значение из двух – он был нашей «целевой переменной» в модели:

1) «0», если СИ прошло прошлую поверку без ремонтов;

2) «1», если СИ не прошло прошлую поверку (прошло только после ремонта).

ГЛАВА 3. СОЗДАНИЕ МОДЕЛЕЙ МАШИННОГО ОБУЧЕНИЯ

Во время экспериментов были рассмотрены и обучены большое количество различных моделей нейронных сетей, которые различались архитектурами и начальными параметрами обучения. Но в данном разделе описаны лишь модели, которые показали реальные результаты, то есть имели хоть какой-то процент правильных прогнозов.

Все модели, которые изучались в процессе разработки подчиняются обобщенной архитектуре, которая представлена на рисунке 10.

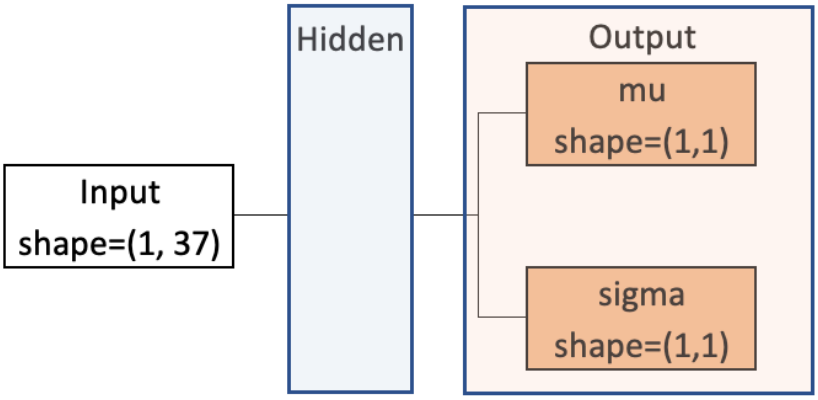

Рис. 10. Обобщенная архитектура моделей

По рисунку выше мы видим, что на вход моделям подаются образцы, которые имеют 37 признаков, что мы подготовили ранее (блок Input), а на выходе моделей мы получаем два значения, которые соответствуют параметрам функции плотности распределения (блок Output) – среднее значение (mu) и стандартное отклонение (sigma).

Блок Hidden соответствует одному или нескольким скрытым полносвязным (dense) слоям или слоям на основе LSTM-ячеек.

В данном разделе представлены:

1) Модели, которые используют только полносвязные слои;
2) Модели, которые используют LSTM-ячейки;
3) Модель на основе логистической регрессии.

Критерий, которому мы стремились – это обоснованный прогноз 90% тестовой выборки. Такой критерий поставили на предприятии исходи из

экономических соображений на основе опыта. Результат считается обоснованным, если модель показывает, что значение погрешности следующего измерения:

1) выйдет за пределы допустимого интервала ([-*relative_error*, +*relative_error*], где *relative_error* - максимальная допустимая относительная погрешность) с вероятностью ≥75% (то есть мы получили вероятность вхождения в интервал <25%), а значение погрешности 20-го измерения будет лежать за пределами допустимого интервала, то считаем, что модель права;

2) будет в пределах допустимого интервала с вероятностью ≥75%, а значение погрешности 20-го измерения будет лежать в пределах допустимого интервала, то считаем, что модель права.

Для каждой рассмотренной модели представлено схематичное описание используемой архитектуры и кривая обучения в виде графика как инструмент визуализации процесса обучения. Данный график иллюстрирует процесс уменьшения потерь (ось Y) при обучении модели с течением времени (эпохи на оси X). В нашем случае, это увеличение вероятности правдоподобия, о направлении (уменьшение) нам говорит знак минус.

Полученные модели тестировались на 3067 образцах. По нашим критериям результат прогнозирования для каждого образца тестовой выборки был определен в одну из 4 групп:

1) true positive – в случае если полученный в результате прогноза результат ≥75%, значение погрешности 20-го измерения лежит в пределах допустимого интервала;

2) false positive – в случае если полученный в результате прогноза результат <25%, значение погрешности 20-го измерения лежит в пределах допустимого интервала;

3) true negative – в случае если полученный в результате прогноза результат <25%, значение погрешности 20-го измерения лежит за пределами допустимого интервала;

4) false negative – в случае если полученный в результате прогноза результат ≥75%, значение погрешности 20-го измерения лежит за пределами допустимого интервала.

Для наглядного представления результатов с учетом установленных критериев использовалась таблица, которая называется матрица ошибок – она объединяет описанные выше значения. Наши матрицы ошибок подчиняются общепринятой, которая представлена на рисунке 11 [16].

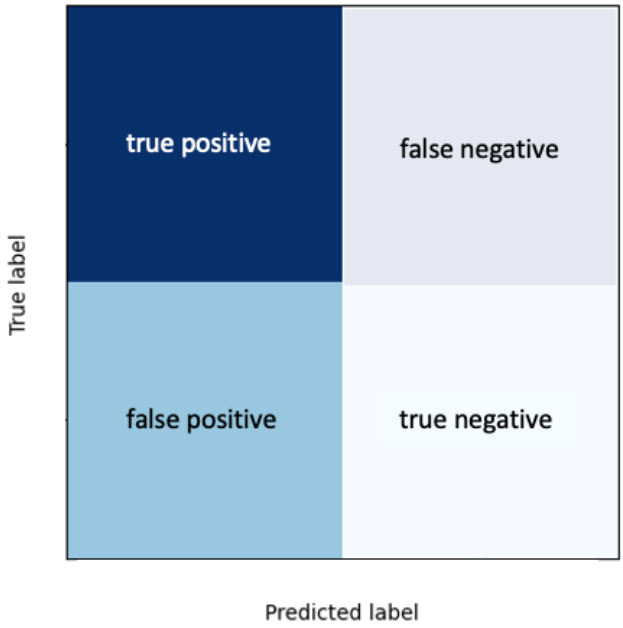

Рис. 11. Шаблон матрицы ошибок

Также были рассмотрены образцы, которые попали в категорию true negative, в рамках работы представлены не все образцы из категории, а только 3 образца, которые имеют низкую, среднюю и близкую к пороговой, вероятность вхождения значения погрешности следующего измерения в допустимый интервал.

3.1. ПОЛНОСВЯЗНЫЕ СКРЫТЫЕ СЛОИ

В полносвязных моделях использовались два скрытых слоя перед разделением (последовательные) для формирования параметров распределения (параллельные). Опытным путем выбранное число юнитов последовательных слоев – 18 и 9. Данная архитектура, представлена на рисунке 12, она использовалась в двух моделях.

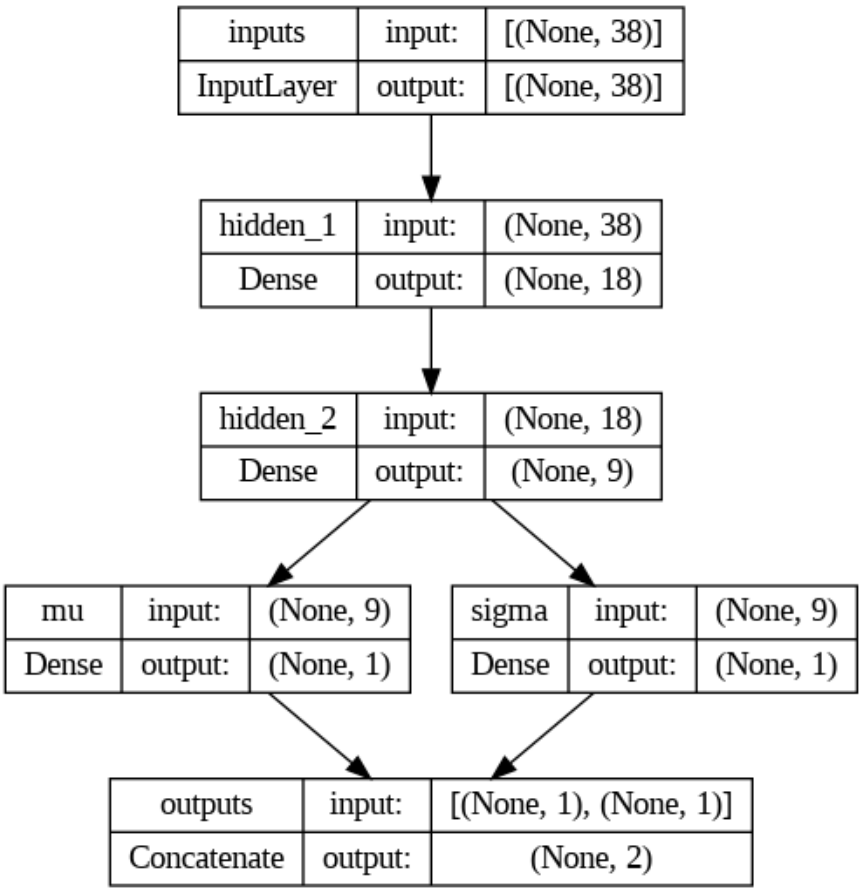

Рис. 12. Схематичное представление архитектуры полносвязных моделей

В последовательных слоях использовалась функция активации, которая называется «Гиперболический тангенс», график зависимости значений функции активации от входного значения представлен на рисунке 13.

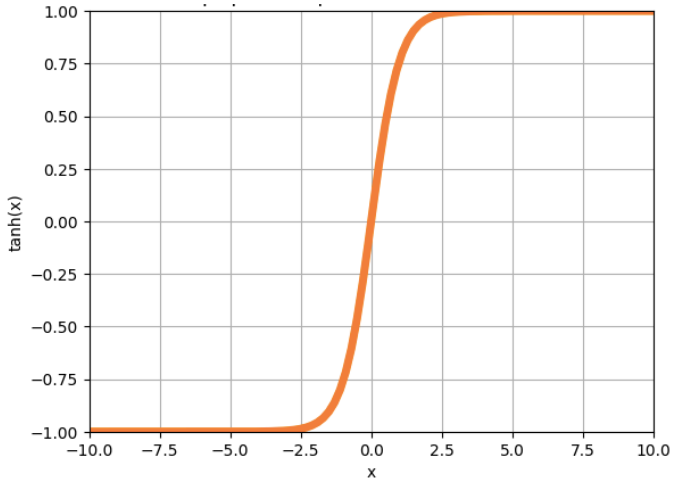

Рис. 13. График функции активации «Гиперболический тангенс»

«Гиперболический тангенс» одна из наиболее популярных функций активации, которые используются в задачах регрессии, она преобразует значения в диапазон [1,-1] по формуле (3.1) [17].

$$\tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}, \quad (3.1)$$

где x – входное значение нейрона.

В обеих моделях использовался оптимизатор «*adam*». «*Adam*» – это оптимизатор адаптивной оценки момента, он направлен на подбор только скорости обучения (остальные параметры являются константами) и содержит преимущества среднеквадратичного распространения («rmsprop») и оптимизатора импульса – высокая эффективность (использует меньше памяти) и подходит для задач с большим количеством параметров. При использовании этого алгоритма оптимизации корректируется только один параметр нейронной сети – скорость обучения, остальные параметры задаются и остаются неизменными [18].

Модели с полносвязными слоями различались тем, что при обучении первой модели использовалась пакетная обработка и данная модель обучалась в течении 500 эпох, а при обучении второй использовалась валидационная выборку и обучалась данная модель в течении 200 эпох.

3.1.1. МОДЕЛЬ С ПАКЕТНОЙ ОБРАБОТКОЙ

Пакетная обработка (от «batch») – это разделение обучающего набора в отдельной эпохе на более мелкие части («пакеты»), обновление модели происходит после обработки одного «пакета». Размер «пакета» был равен 256. В обучающем наборе 7155 образцов, соответственно, модель обновила свои параметры 14000 раз.

У модели 893 параметра обучения. Значение логарифма функции потерь на тренировочной выборке на конец обучения: -4.0852. На рисунке 14 представлена кривая обучения модели.

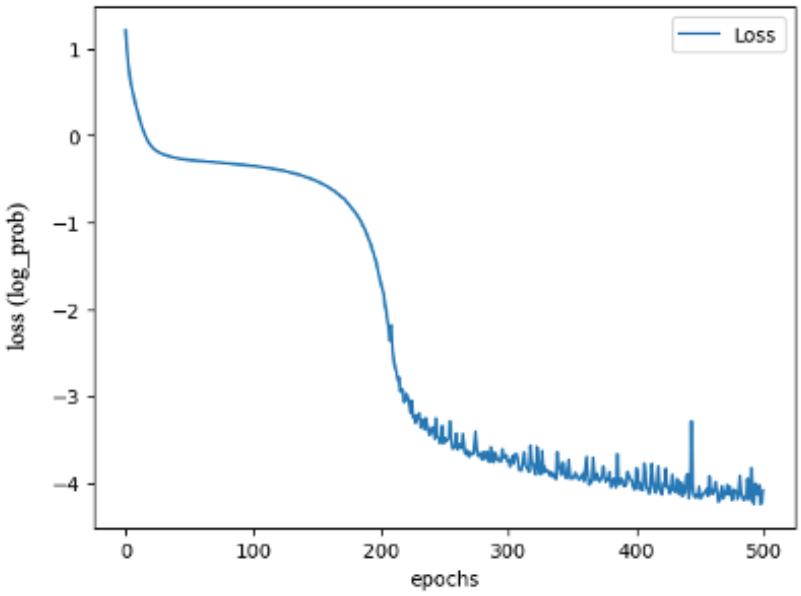

Рис. 14. Кривая обучения модели с пакетной обработкой

По рисунку мы видим, что значение функции потерь, в целом, сходится, но кривая обучения имеет некоторый горб, которые трудно объяснить сразу, он требует более углублённого анализа.

3.1.2. МОДЕЛЬ С ВАЛИДАЦИОННОЙ ВЫБОРКОЙ

Эпоха в глубоком обучении – это одна итерация по обучающему набору данных. Одна эпоха завершена, когда модель обработала весь набор данных и обновила свои параметры на основе алгоритма оптимизации и значения функции потерь. То есть, в данном случае, модель обновила свои параметры 200 раз.

Валидационная выборка – это часть обучающего набора, которая используется для проверки модели. Библиотека TensorFlow позволяет делать это «на лету» и корректировать поведение модели, чтобы предотвратить переобучение.

У модели 893 параметра обучения. Значение логарифма функции потерь на тренировочной выборке на конец обучения: -3,6106, на валидационной: -3,7323.

На рисунке 15 представлена кривая обучения модели. Несмотря на то, что это «вероятность», она может быть больше 1, так как выражена в логарифмическом эквиваленте.

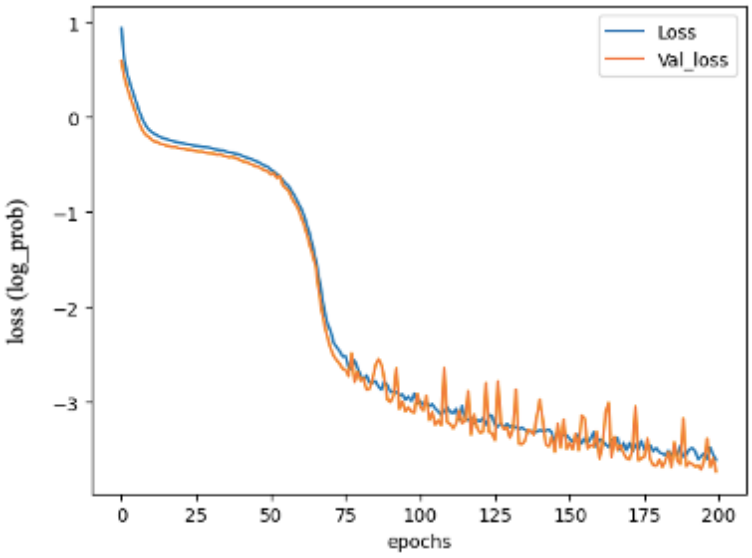

Рис. 15. Кривая обучения модели с валидационной выборкой

Кривая обучения показывает, что модель приближенно описывает как тренировочные данные, так и новые [19]. Мы получили кривую обучения, которая похоже на ту, что представлена выше (для модели с пакетной обработкой). Можно сделать вывод, что наличие валидационной выборки в данной задаче повышает эффективность, так как данная модель обучалась быстрее, чтобы получить результат, который практически не отличается от результата предыдущей модели.

Так как модель с валидационной выборкой эффективнее, то будем рассматривать результаты для нее.

На рисунке 16 представлена матрица ошибок, согласно требованиям выше.

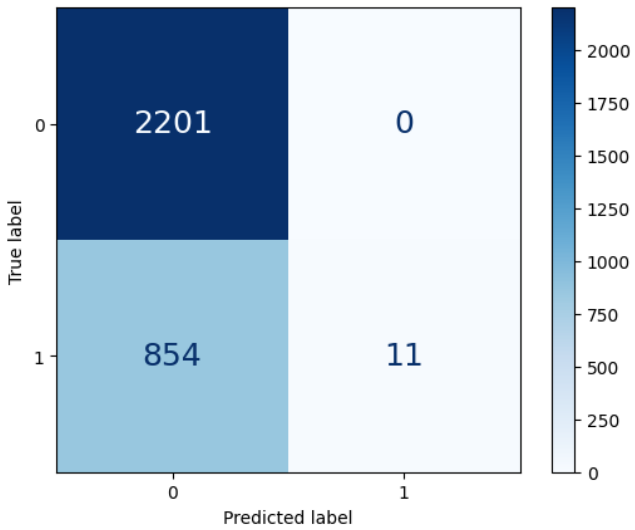

Рис. 16. Матрица ошибок

На осях 0 – означает, что значение погрешности 20-го измерения будет лежать за пределами допустимого интервала, а 1 – наоборот.

Довольно плохие результаты, 854 «ошибки». В данной задаче, трудно определить, что такое точность модели, поэтому будем считать описанные выше требования критериями точности. Соответственно, точность данной модели 72,2%.

Рассмотрим образцы, которые не подходят под критерии, для этого построим графики, которые иллюстрируют динамику погрешности с каждым измерением. Они представлены на рисунке 17.

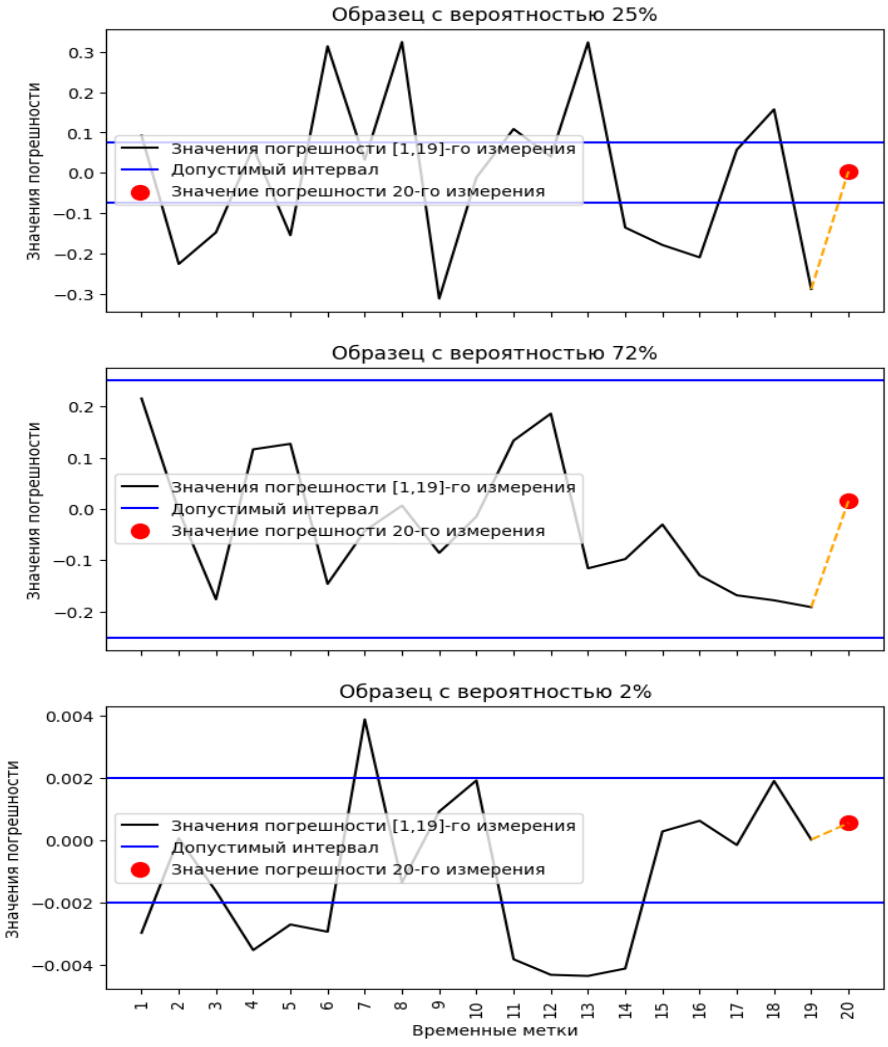

Рис. 17. График динамики значений погрешности

Образец с вероятностью 72% не вошел в «правильные», но по графику можно сказать, что такой результат обоснован, так как образец имеет несколько точек близких к пороговому значению.

Странные результаты для 1-го и 3-го образца, скорее всего такая вероятность ошибочна. Но если у 3 образца, такой результат можно обосновать очень малым диапазоном допустимых значений, то первый образец не имеет видимых обоснований.

3.2. МОДЕЛИ, КОТОРЫЕ ИСПОЛЬЗУЮТ LSTM-ЯЧЕЙКИ

3.2.1. МОДЕЛЬ С ОДНИМ LSTM-СЛОЕМ

В LSTM-слое использовалось 10 нейронов с функцией активации «Гиперболический тангенс». Архитектура данной нейронной сети схематично представлена на рисунке 18.

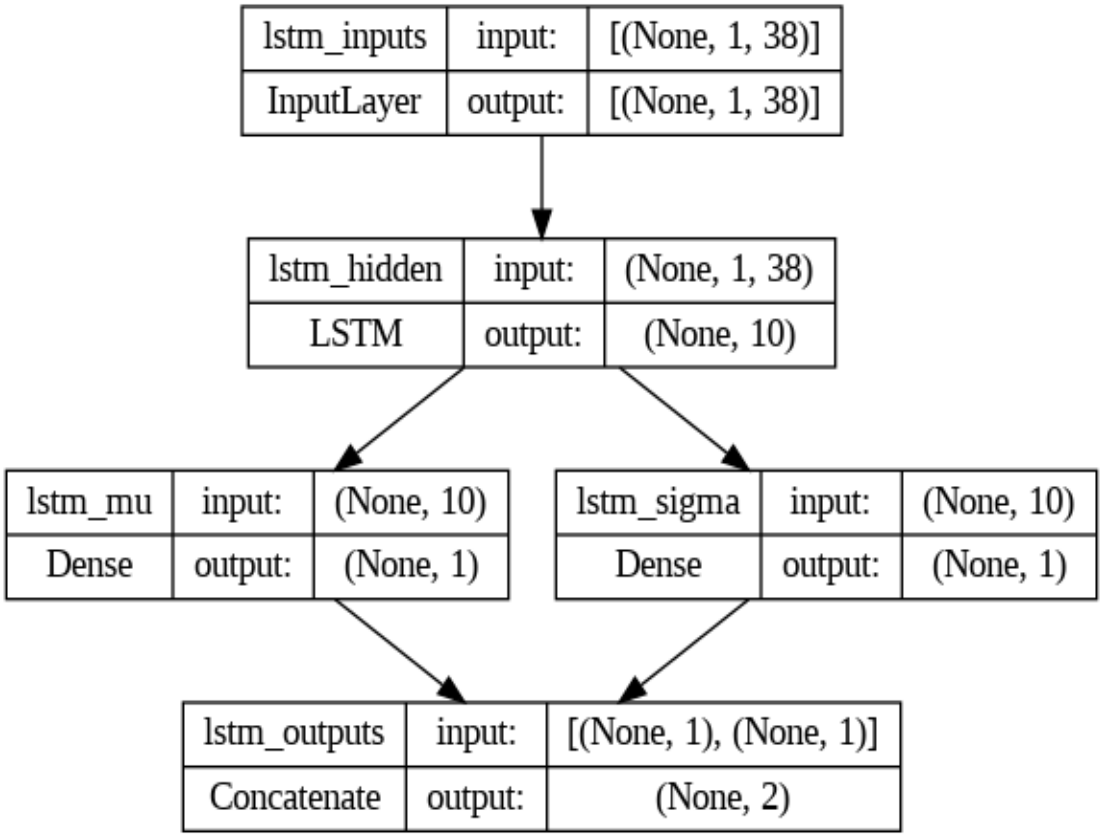

Рис. 18. Архитектура модели с одним LSTM-слоем

Процесс обучения проходил с использованием оптимизатора «*SGD*» и валидационной выборки.

«*SGD*» – оптимизатор стохастического градиентного спуска, его главное преимущество – высокая частота обновления параметров модели [18]. С помощью данного алгоритма мы будто «шагаем» в сторону самой низкой точки.

У модели 1982 параметра обучения. Значение логарифма функции потерь на тренировочной выборке на конец обучения: -0,3067, на валидационной: -0,2008. На рисунке 19 представлена кривая обучения модели.

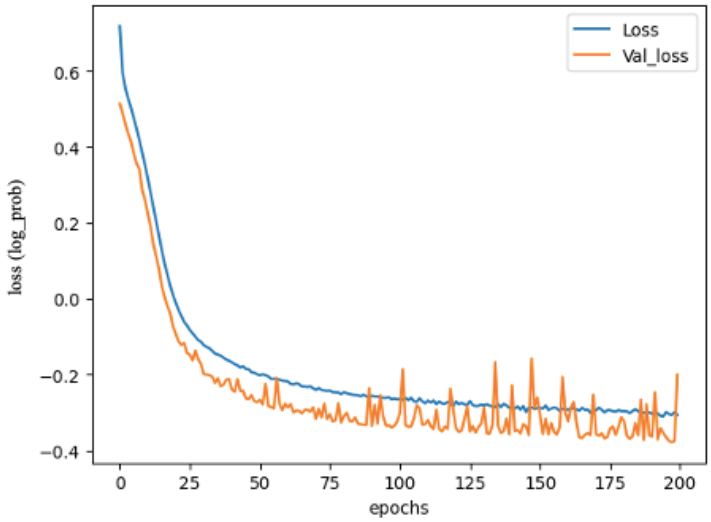

Рис. 19. Кривая обучения LSTM-модели с оптимизатором «SGD»

Это единственная модель, которая не имеет горба на кривой обучения. По графику можно предположить, что модель довольно хорошо описывает данные, так как, значения функции потерь на графике сходятся. Чтобы убедится в этом мы построили матрицу ошибок, она представлена на рисунке 20.

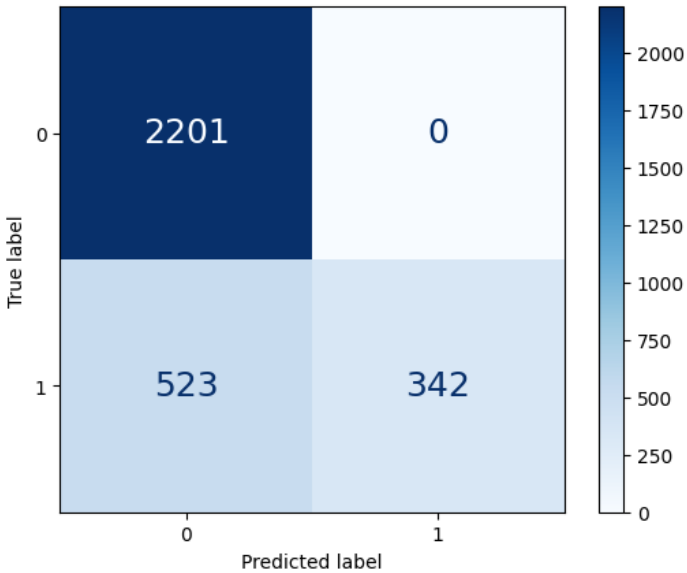

Рис. 20. Матрица ошибок

Матрица ошибок показывает, что LSTM-ячейки помогли улучшить точность результата, так как они лучше учитывают временные

последовательности. Данная модель совершила существенно меньше «ошибок» – ее точность 82%. Рассмотри некоторые образцы, прогнозная погрешность которых не подходит под критерии точности. 3 образца представлены на рисунке 21.

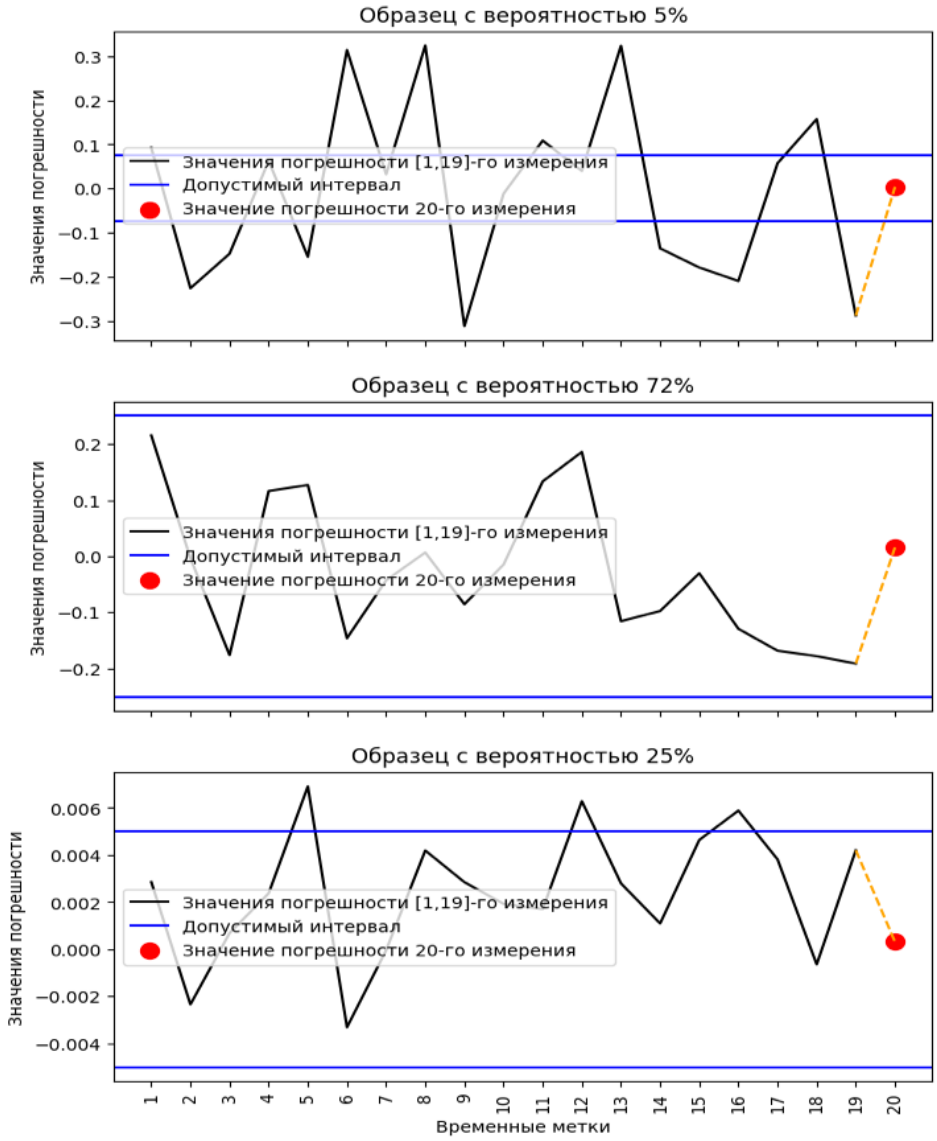

Рис. 21. График динамики значений погрешности

По графику динамики значений погрешности для этих образцов, трудно сказать, что модель ошиблась, но это субъективное мнение. Требуется посмотреть другие признаки. Например, 3-й образец с 3 месторождения с 17 площадки типа «Метран» производства России. Другие «похожие» образцы с такими же значениями признаков также имеют прогнозные значения вероятности в диапазоне 20-40%. Это может означать, что модель не ошиблась с результатом для данного образца, но он требует человеческого вмешательства.

3.2.2. МОДЕЛЬ СО СКРЫТЫМ BILSTM-СЛОЕМ

В целях улучшения «точности» модели была разработана более сложная архитектура с использованием LSTM-слоя, который «проходит» последовательность признаков в обоих направлениях и добавления по одному скрытому полносвязному слою для параметров (после разделения потока) [20].

В двунаправленном LSTM-слое использовалась нормализованная инициализация весов. При такой инициализации начальные веса присваиваются на основе значений нормального распределения.

Так как архитектура довольно нагруженная, то использовался метод борьбы с переобучением модели – «dropout».

На рисунке 22 схематично представлена архитектура полученной модели.

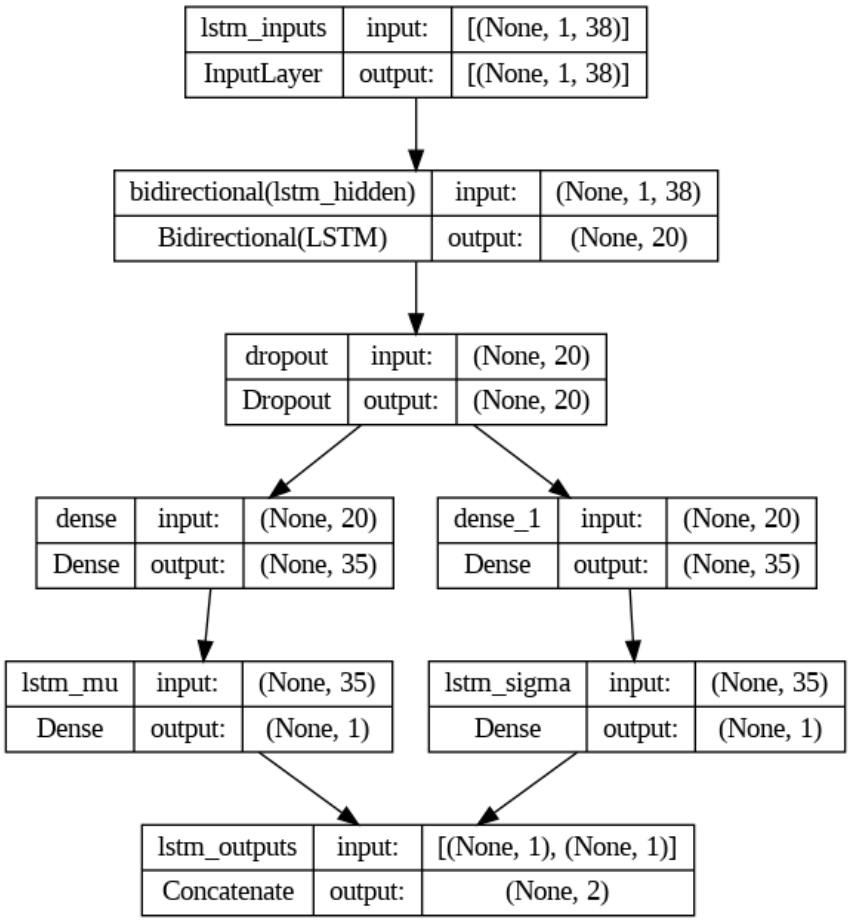

Рис. 22. Схематичное представление архитектуры BiLSTM-модели

Переобученная модель – это такая модель, которая вместо обучения адаптируется к данным, то есть модель не «понимает реальность» и не может правильно обрабатывать незнакомые образцы. «Dropout» (метод исключения) –

метод борьбы с переобучением модели нейронной сети, суть которого это исключение нейронов с заданной вероятностью *p* (исключаем *p*% нейронов) [21]. Таким образом модель не теряет способность к обобщению.

У модели 5462 параметра обучения. При обучении использовался оптимизатор «*adam*». На рисунке 23 представлена кривая обучения модели.

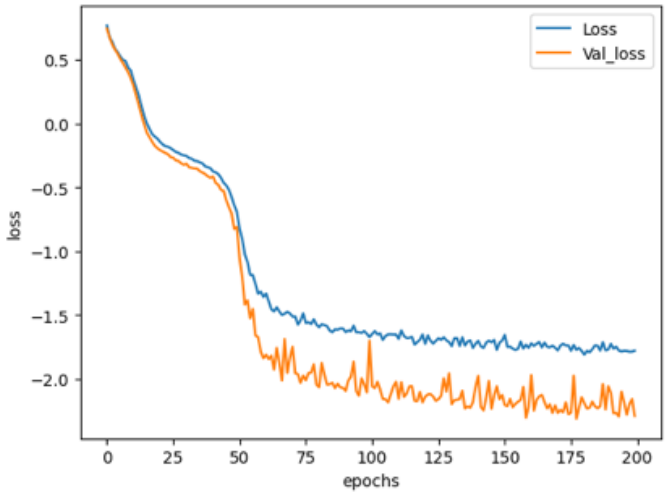

Рис. 23. Кривая обучения BiLSTM-модели

Кривая показывает, что на валидационной выборке значение функции потерь сильно ниже в конце обучения, это может означать, что набор данных для проверки не отражает всю реальность, скорее всего он больше похож на тренировочный. Возможно, модель переобучена, поэтому возьмем значения функции потерь для 50 эпох: -1,1924, на валидационной: -1,260. На рисунке 24 представлена матрица ошибок для данной модели.

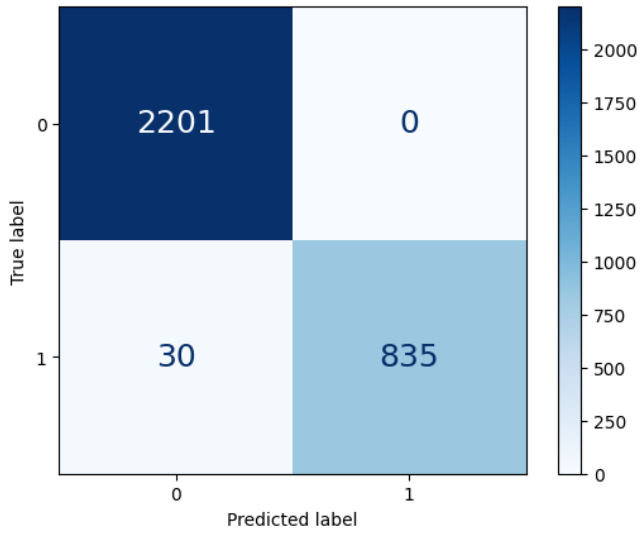

Рис. 24. Матрица ошибок

По ней мы видим, что модель хорошо оценила данные согласно условиям выше. Точность модели – 98%. Но также стоит рассмотреть «ошибочные» образцы более подробно, так как понятие точности для моделей генерации распределения очень размыто. На рисунке 25 представлены графики динамики значений погрешности для этих образцов.

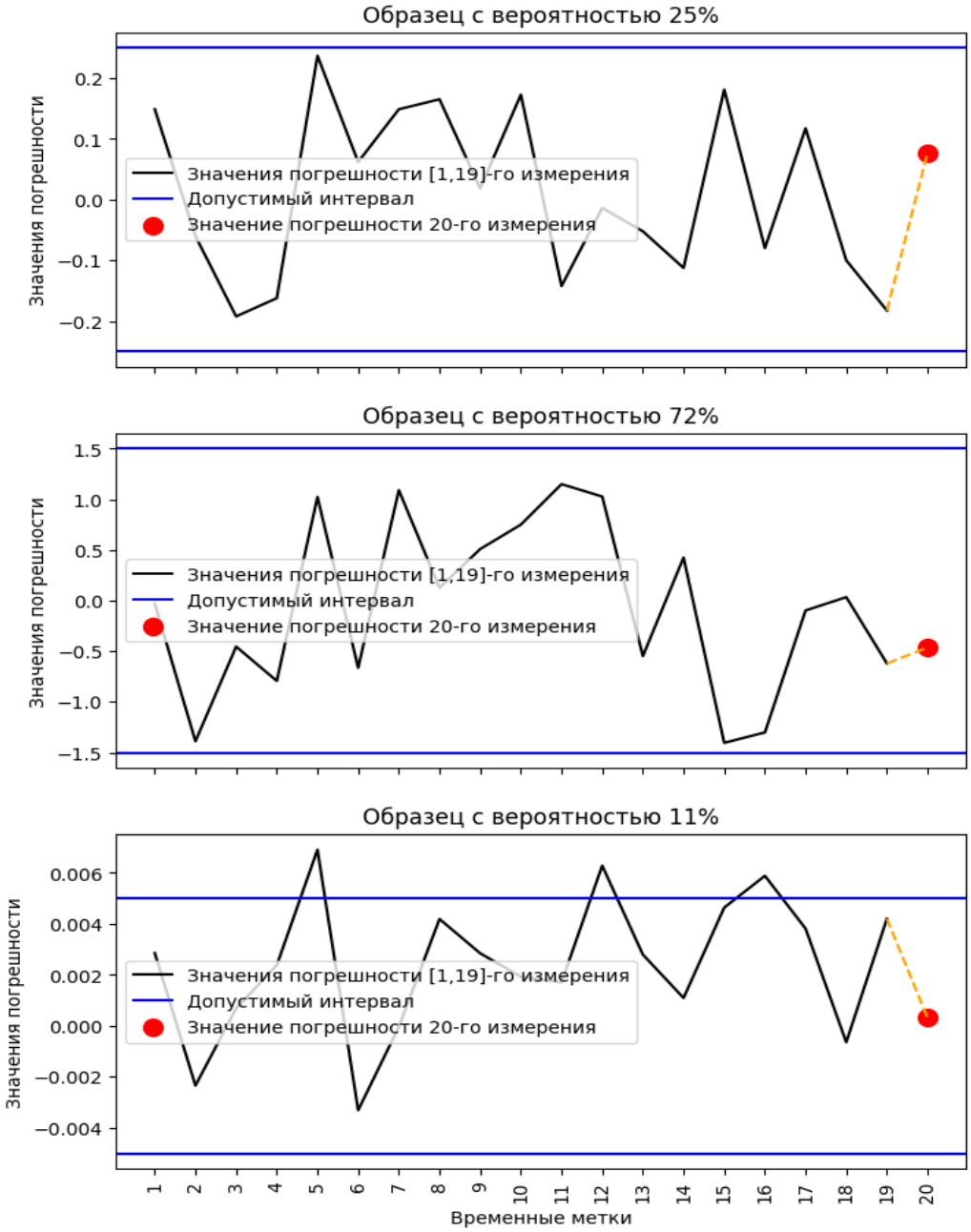

Рис. 25. Тестовые образцы

Все образцы имеют большое количество значений, которые близки к допустимым, поэтому эти графики можно считать обоснованием ответа.

3.3. РЕЗУЛЬТАТЫ ГЛАВЫ

По данной главе мы получили два основных результата:

1) Модель на основе логистической регрессии не обучилась. Ее полученные значения равны «0» для тестовой выборки. Хотя при сравнении с правильными значениями точность модели >70%, нельзя утверждать, что данный результат допустим, так как в выборке >70% «нулей». В связи с таким результатом можно сделать вывод, что в текущей задаче необходимо использовать именно глубокое обучение.

2) Мы отдали предпочтение модели нейронной сети с двунаправленным LSTM-слоем в основе архитектуры и четырьмя параллельными параметрическими слоями (по два для каждого получаемого параметра), так как она показала лучшие результаты, согласно матрице ошибок. Возможно, данная модель все-таки была переобучена, но мы взяли промежуточный результат (50 эпох), который удовлетворяем поставленным требованиям. Исходный код реализации данной архитектуры модели нейронной сети на языке программирования Python с использованием библиотеки глубокого обучения keras представлен в приложении 1.

ГЛАВА 4. ИССЛЕДОВАНИЕ ВЛИЯНИЯ ОТДЕЛЬНЫХ ПРИЗНАКОВ НА РЕЗУЛЬТАТ ПРОГНОЗИРОВАНИЯ

4.1. ОПИСАНИЕ МЕТОДА

Чтобы определить силу и характер влияния признаков на полученные результаты модели, мы использовали значения Шепли (вектор SHAP).

Вектор Шепли – это оптимальное распределение выигрыша между игроками в задачах теории кооперативных игр. [22] Данный метод был назван в честь американского математика Ллойда Шепли.

Согласно теореме Шепли, для любой игры существует единственное распределение выигрыша (вектор Шепли), которое удовлетворяет следующим аксиомам:

1) Линейность – полезность игрока в двух играх получается сложением полезностей в каждой;

2) Симметричность – полезность игрока не зависит от его положения;

3) Эффективность – полное распределение имеющегося «результата игры»;

4) Аксиома бесполезного игрока – бесполезный игрок не получает ничего.

Для оценки важности того или иного признака по данному методу мы рассчитывали значения Шепли, которые являются разницей между предсказаниями модели «с» и «без» конкретного признака. Если же строить аналогию по теории игр (откуда пришли значения Шепли), то признаки – это игроки, предсказания – результат игры, а оцениваем вклад каждого игрока в результат игры. Значения Шепли рассчитываются по формуле (4.1) [22].

$$\phi_i(p) = \sum_{S \subseteq \frac{n}{\{i\}}} \frac{|S|!(n-|S|-1)!}{n!} (p(S \cup \{i\}) - p(S)), \qquad (4.1)$$

где S – набор признаков без i-того признака,

n – всего признаков в наборе,

$p(S)$ – это предсказания модели без i-того признака,

$p(S \cup \{i\})$ – это предсказания модели с i-тым признаком.

4.2. ПРИМЕНЕНИЕ МЕТОДА

Схематично процесс расчета значений Шепли можно представить на рисунке 26.

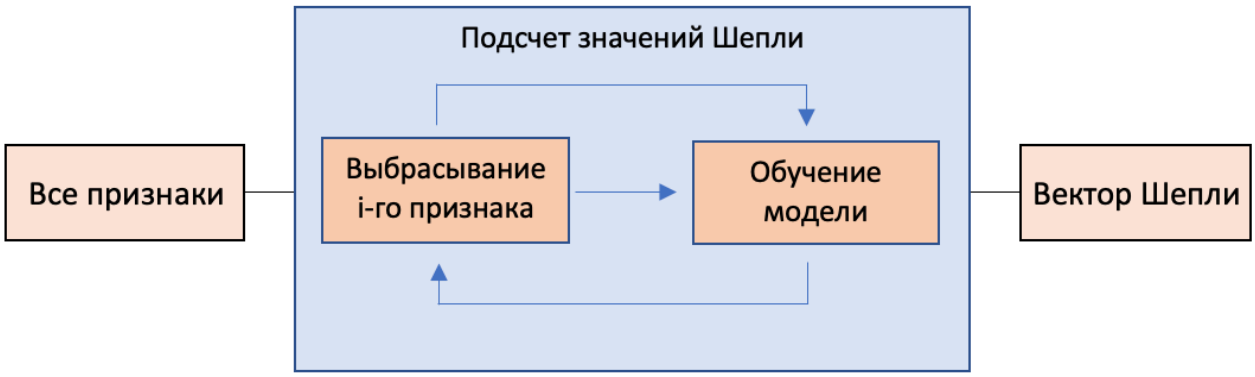

Рис. 26. Процесс получения вектора Шепли

Схема показывает, что для всех образцов, мы исключали один из признаков, затем обучали модель на полученном наборе и делали прогноз для этого образца. Затем полученный результат сравнивался с результатом прогноза модели, которая обучалась на всех признаках. Так как данные вычисления очень требовательны к ресурсам (необходимо обучить модель на всех возможных подмножествах признаков, что чаще всего невыполнимо), а значения погрешности измерений являются основными признаками, то в данном случае мы рассматривали только описательные признаки.

С использованием библиотеки shap на основе посчитанных значений была построена диаграмма значений Шепли, которая представлена на рисунке 27. Цвет показывает значение признака. Чем толще линия, тем большее количество образцов имеет советующее значение Шепли. Признаки расположены по уменьшению влияния.

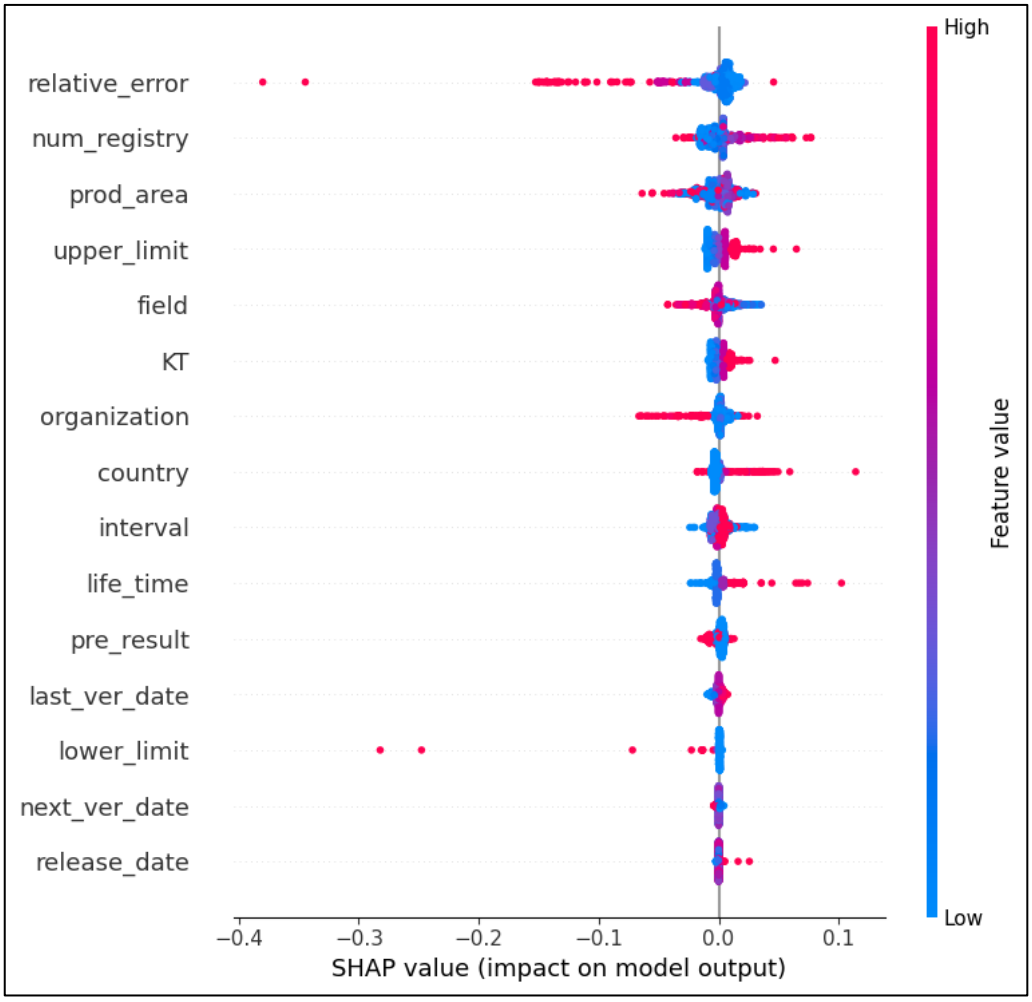

Рис. 27. Диаграмма значений Шепли

Так на примере признака *relative_error*: чем меньше значение максимальной допустимой погрешности, тем меньше вероятность, что реальное значение будет принадлежать найденному распределению, что соответствует действительности.

Также мы видим, что признак следующей даты поверки (*next_ver_*date) практически не оказывает никакого влияния на результаты модели.

В общем случае можно сделать вывод, что данные признаки слабо, но все же влияют на прогнозы модели, так как значения Шепли находятся в диапазоне [-0.4, 0.2].

ГЛАВА 5. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ИНСТРУМЕНТА

Всё, что описано ниже представлено в обобщенном виде, так являются частью системы компании и не подлежит разглашению.

5.1. АРХИТЕКТУРА

Архитектура современных информационных систем с использованием алгоритмов машинного обучения содержит множество сервисов, которые взаимодействую между собой и выстраиваются в процесс взаимодействия пользователя с моделями машинного обучения.

В нашем инструменте точкой входа является интерфейс пользователя в виде веб-приложения, которое представляет пользователю «сырые» данные в форме таблиц.

Инструмент содержит следующие компоненты:

1) Веб-интерфейс;
2) База данных;
3) Модуль подготовки данных;
4) Модуль сбора контрольных данных.

Данные компоненты взаимодействую между собой с помощью API-запросов. Обобщенная архитектура представлена на рисунке 28.

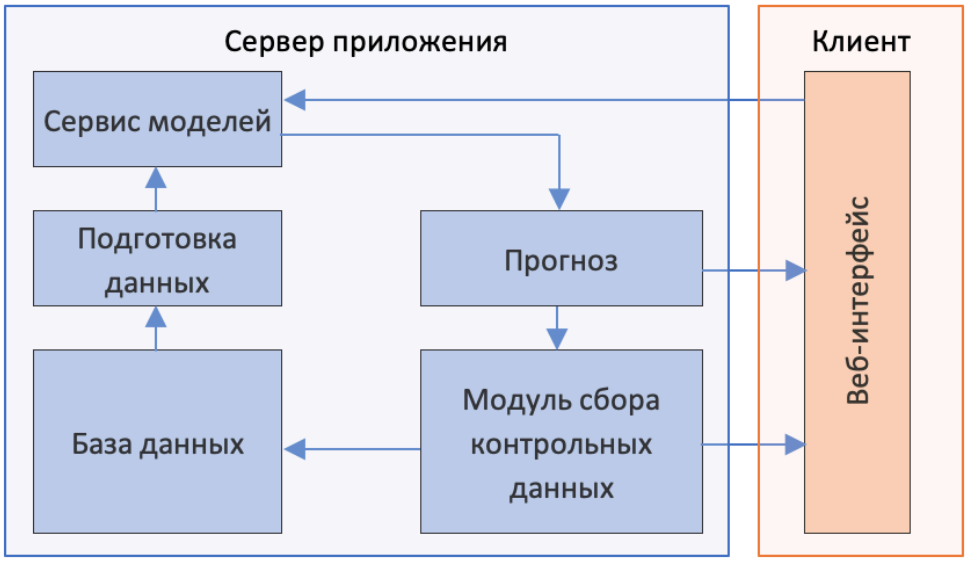

Рис. 28. Архитектура инструмента

В базе данных хранятся исходные данные, с которыми работает пользователь в системе. Модуль подготовки данных отвечает за сбор

дополнительных данных, кодирование существующих и создание новых признаков. Сервис моделей предоставляет актуальную модель для получения прогноза. Полученный прогноз направляется пользователю, а также в модуль сбора контрольных данных. Модуль сбора контрольных данных необходим для контроля и оценки модели. Контроль представляет собой показатели точности и мониторинг производительности, данные отправляются в интерфейс пользователя для оценки администраторами системы. Такие данные обычно представляются в виде дашбордов. Оценка модели необходима для дообучения модели, после наполнения набора данных с течением времени.

5.2. МОДУЛЬ ПОДГОТОВКИ МОДЕЛЕЙ

Модуль подготовки моделей реализован в виде flask-приложения (используется python-библиотека Flask). Приложение содержит 3 функции:

1) build_model – загрузка обученной модели;

2) predict – создание прогноза (конечная точка «*http://<адрес_сервера_модели>/predict*»);

3) fit – обучение (конечная точка «*http://<адрес_сервера_модели>/fit*»).

Обученная модель сохранена с помощью метода *save_weights* класса *Model* python-библиотеки *Tensorflow* в файл с расширением .h5. Такой формат имеют HDF5 (иерархический формат данных) файлы, выбран данный формат, потому что веса больших моделей содержат много цифровой информации, а данный формат позволяет хранить ее эффективно. При вызове функции *build_model* сначала создается архитектура модели, затем загружаются веса обученной модели из .h5-файла с помощью метода *load_weights* того же класса с указанием пути до файла в качестве аргумента и возвращается эта модель.

Когда пользователь открывает веб-приложение, отправляется HTTP GET-запрос на конечную точку «*http://<адрес_сервера_модели>/predict*» модуля подготовки моделей. При получении запроса на данную конечную точку в модуле подготовки моделей вызывается функция *predict*:

1) сначала мы получаем данные, для которых необходимо сделать прогноз из модуля подготовки данных с помощью HTTP GET-запроса;

2) получаем модель с помощью вызова функции *build_model*;

3) затем обращаемся к методу *predict* полученной модели (получена модель является экземпляром класса *Model*).

Запрос возвращает JSON-данные в виде, который представлен на рисунке 29. Для этого используется python-библиотека json.

```
{
    "data": [
        {
            "id": "<ID_оборудования>",
            "value": "<Значение_вероятности>"
        },
        ...
    ],
    "dt_predict": "<Дата_прогноза>"
}
```

Рис. 29. Структура ответа на конечную точку «/*predict*»

В ответе содержатся значения прогноза и id оборудования, а также дата создания прогноза. Данные прогноза оправляются в интерфейс пользователю, а также в модуль сбора контрольных данных для преобразования и записи в базу данных. В дальнейшем эти данные будут использоваться для изучения различных зависимостей. Эти данные представляются в виде графиков и диаграмм в интерфейсе администраторов системы.

Запросы к конечной точке «*http://<адрес_сервера_модели>/fit*» отправляются из интерфейса администратора системы. Человек, которые обладает данными правами оценивает модель с точки зрения бизнеса, например, на основе матрицы ошибок, которые были описаны в главе 3 (с течением времени). При получении запроса на данную конечную точку вызывается функция *fit*:

1) сначала мы получаем данные, на которых необходимо дообучить модель из модуля подготовки данных с помощью HTTP GET-запроса;

2) получаем модель с помощью вызова функции build_model;

3) затем обращаемся к методу *fit* полученной модели (получена модель является экземпляром класса Model).

ГЛАВА 6. ОЦЕНКА ЭФФЕКТИВНОСТИ ПРИМЕНЕНИЯ ИНСТРУМЕНТА

Главной целью данной главы является теоретическая оценка эффективности применения разработанного функционала. Для выполнения данной цели нам нужно:

1) определить, что мы будем использовать как показатель эффективности;

2) рассчитать экономическую эффективность от внедрения разработанного функционала в общую систему организации;

3) сделать выводы о рентабельности/окупаемости проекта.

Обычно, при расчете стоимости поверки в бюджет закладывается стоимость ремонта около 70% оборудования. В итоге часть этих денег действительно тратится на ремонт, часть на покупку нового оборудования, а часть остается не использована. К тому же, при отправке оборудования в поверку ожидаемые сроки составляются из сроков поверки, непосредственно, и сроков транспортировки оборудования. Если при этом прибор не пройдет поверку, то его направят на ремонт (добавляется срок ремонта). Либо, если оборудование не подлежит ремонту, то необходимо его списать и закупить новое, что тоже добавляет срок. Соответственно, после ремонта/замены оборудования, нужно все равно провести поверку оборудования.

В данном случае под эффективностью мы будем рассматривать разницу между затратами на поверку с заложенной стоимостью ремонта оборудования и затратами на поверку с заложенной стоимостью ремонта оборудования с низкой вероятностью прохождения поверки (<25%).

Рассчитаем эффективность на примере датчика давления Метран-150 с номером государственного реестра 32854-13 при условии проведения поверки/ремонта в ФБУ «Тюменский ЦСМ». В нашем наборе данных 2453 таких датчика.

Стоимость поверки такого датчика 3245 рублей. Стоимость ремонта может варьироваться в зависимости от причины неисправности и может составлять от 1000 до 20000. Стоимость такого нового датчика составляет от 35000 рублей.

Рассчитываем бюджет по формуле (5.1):

$$Б = n * \left(cost_p + \overline{cost_r}\right) \quad (5.1)$$

где n – количество оборудования, шт.,

$cost_p$ – стоимость поверки оборудования,

$\overline{cost_r}$ – средняя стоимость ремонта оборудования.

Итого для проведения поверки мы закладываем бюджет по формуле 5.1 равный – ~32 489 985.

Теперь посмотрим на вероятность прохождения поверки, прогноз которой мы получили с помощью нашего инструмента. Мы проиллюстрировали результат с помощью гистограммы на рисунке 30.

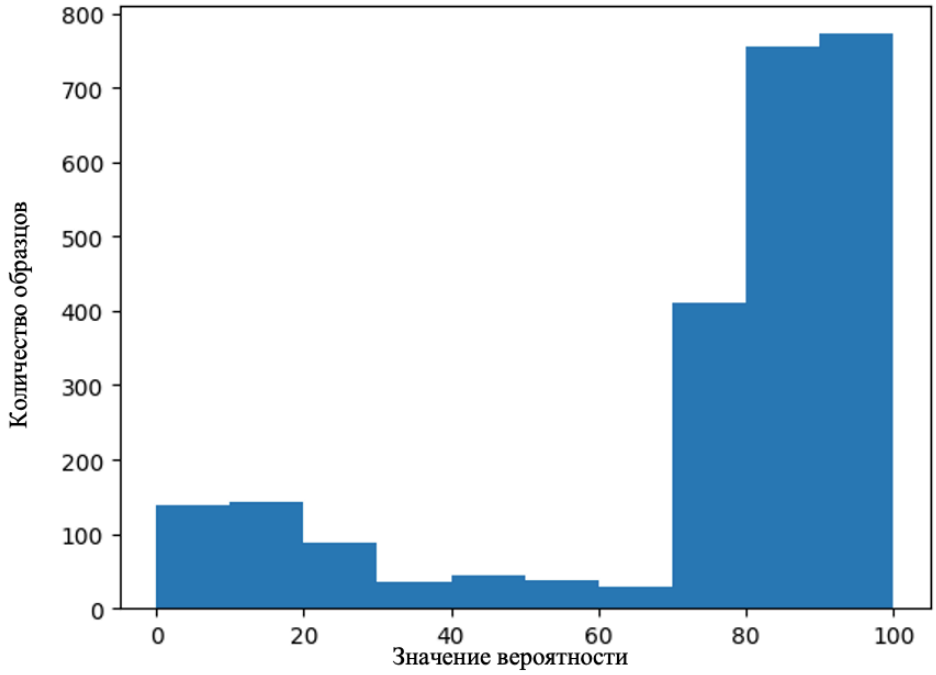

Рис. 30. Гистограмма значений вероятности для Метран-150

По рисунку мы видим, что:

1) 1914 приборов получили вероятность >75%;
2) 357 приборов получили вероятность <25%.

Изменим формулу 5.1 расчета бюджета для нашего случая (5.2):

$$Б = n_{75} * cost_p + n_{25} * \left(cost_p + \overline{cost_r}\right) \quad (5.2)$$

где n_{75} – количество приборов, которые получили вероятность >75%, шт.,

n_{25} – количество приборов, которые получили вероятность ≤25%, шт.

Но так как все, что попало в интервал [25, 75] %, требует оценки со стороны человека, то мы включаем это количество в n_{25}. По формуле 5.2 мы получили сумму равную 13 349 985.

Таким образом высвобождение бюджета составит 41%.

ЗАКЛЮЧЕНИЕ

В настоящей работе были исследованы математические методы, которые позволяют оценить вероятность прохождения поверки средством измерения. На основе этих методов была сформулирована концепция такой оценки и построены несколько моделей машинного обучения и выбрана та, что лучше удовлетворяет поставленным критериям отбора – она имеет «точность» 92%.

Результаты данной работы являются новыми, так как мы не нашли в литературе упоминания использования методов машинного обучения или похожей концепции оценки в области вероятностной оценке результатов поверки.

Полученная модель ляжет в основу разработки программного инструмента, который будет применяться на предприятии. Поэтому была оценена экономическая эффективность применения такого инструмента. Полученные результаты оценивания могут использоваться руководством для:

1) принятия координационных решений руководством и руководителями проектов в рамках данной организации;

2) планирования корректирующих и предупреждающих действий в области обеспечения единства измерений.

Проблема разработанной концепции заключается в том, что основой для обучающего набора данных являются измерения в течении межповерочного интервала. Так как существуют средства измерения с очень большим межповерочным интервалом, например, 5-10 лет, то собрать набор данных для таких приборов очень сложно. Допущение состоит в том, что чаще всего такие средства измерений имеют низкую стоимость поверки/ремонта или замены, потому что не являются высокоточными, как пример можно привести обычные счетчики воды в жилых помещениях.

Результаты, полученные в данной работе, будет применятся при развитии темы изучения способов прогноза остаточного ресурса оборудования.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Федеральный закон от 26.06.2008 № 102-ФЗ «Об обеспечении единства измерений». КонсультантПлюс. – URL: https://www.consultant.ru/document/cons_doc_LAW_77904/ (дата обращения: 12.05.2023). – Текст: электронный.

2. ПР 50.2.006-94. Правила по метрологии. Государственная система обеспечения единства измерений. Порядок проведения поверки средств измерений. издание официальное: утвержден и введен в действие Приказом Федерального агентства по техническому регулированию и метрологии от 18.07.1994 г. № 125 / разработан Всероссийским научно-исследовательским институтом метрологической службы Госстандарта России. – Москва: Стандартинформ, 2017. – 43 с. – Текст: непосредственный.

3. Федеральный закон от 27.12.2002 № 184-ФЗ «О техническом регулировании». КонсультантПлюс. – URL: https://www.consultant.ru/document/cons_doc_LAW_40241/ (дата обращения: 12.05.2023). – Текст: электронный.

4. Kanter, J. M. Deep feature synthesis: Towards automating data science endeavors / J. M. Kanter, K. Veeramachaneni. – IEEE International Conference on Data Science and Advanced Analytics (DSAA), Paris, 2015. – 10 с.

5. Pearson correlation coefficient. Википедия. Свободная энциклопедия. – URL: https://en.wikipedia.org/wiki/Pearson_correlation_coefficient (дата обращения: 17.03.2023). – Текст: электронный.

6. Casella, G. Statistical inference. Vol. 2. / G. Casella, R. L Berger. – Duxbury Pacific Grove, CA, 2002. – 686 с. – Текст: непосредственный.

7. Q-Q Plots. Хабрахабр. Экосистема для развития людей, вовлеченных в IT. – URL: https://habr.com/ru/post/578754/ (дата обращения: 16.03.2023). – Текст: электронный

8. PDF is not a probability. Medium. – URL: https://towardsdatascience.com/pdf-is-not-a-probability-5a4b8a5d9531 (дата обращения: 15.03.2023). – Текст: электронный

9. Bishop, C. M. Mixture Density Networks / C. M. Bishop. – Aston University, Brimingham, U.K., 1994. – 26 с. – URL: https://publications.aston.ac.uk/id/eprint/373/1/NCRG_94_004.pdf (дата обращения: 12.05.2023). – Текст: электронный.

10. Вентцель, Е. С. Теория вероятностей / Е. С. Вентцель. – 10-е изд. – М.: Высш. Шк., 2006. – 575 с. – Текст: непосредственный.

11. Clevert, D. A. Fast and Accurate Deep Network Learning by Exponential Linear Units / D. A. Clevert, T. Unterthiner, S. Hochreiter. – ArXiv. – URL: https://arxiv.org/abs/1511.07289 (дата обращения: 15.04.2023). – Текст: электронный.

12. Agrawal, S. Hyperparameters in Deep Learning / S. Agrawal. – Medium. – URL: https://towardsdatascience.com/hyperparameters-in-deep-learning-927f7b2084dd (дата обращения: 25.04.2023). – Текст: электронный.

13. Osborne, M. R. Least squares and maximum likelihood / M. R. Osborne. – URL: https://maths-people.anu.edu.au/~mike/lsnml.pdf (дата обращения: 10.02.2023). – Текст: электронный.

14. Hochreite, S. Long short-term memory / S. Hochreite, J. Schmidhuber. – Neural Computation, 1997. – 55 с. – URL: https://www.researchgate.net/publication/13853244_Long_Short-term_Memory (дата обращения: 10.02.2023). – Текст: электронный.

15. Как легко понять логистическую регрессию. Хабрахабр. Экосистема для развития людей, вовлеченных в IT. – URL: https://habr.com/ru/companies/io/articles/265007/ (дата обращения: 12.05.2023). – Текст: электронный.

16. Fawcett, T. An Introduction to ROC Analysis / T. Fawcett. – Institute for the Study of Learning and Expertise, USA, 2006. – 14 с. – URL: https://people.inf.elte.hu/kiss/13dwhdm/roc.pdf (дата обращения: 12.05.2023). – Текст: электронный.

17. Выбор слоя активации в нейронных сетях: как правильно выбрать для вашей задачи. Хабрахабр. Экосистема для развития людей, вовлеченных в

IT. – URL: https://habr.com/ru/articles/727506/ (дата обращения: 12.05.2023). – Текст: электронный.

18. Murphy, K. P. Probabilistic Machine Learning: An Introduction. / K. P. Murphy – MIT Press, 2022. – 864 с. – Текст: непосредственный.

19. Claude, S. Encyclopedia of Machine Learning / S. Claude, G. I. Webb. – Springer. – 578 с. – Текст: непосредственный.

20. Schuster, M. Bidirectional recurrent neural networks / M. Schuster, K. K. Paliwal – B: IEEE Transactions on Signal Processing 45.11, 1997. – 9 с. – URL: https://maxwell.ict.griffith.edu.au/spl/publications/papers/ieeesp97 _schuster.pdf (дата обращения: 12.05.2023). – Текст: электронный.

21. Dropout: A Simple Way to Prevent Neural Networks from Overfitting / N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov. – Journal of Machine Learning Research, 2014. – 30 с. – URL: https://jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf (дата обращения: 12.05.2023). – Текст: электронный.

22. Aas, K. Explaining individual predictions when features are dependent: more accurate approximations to shapley values / K. Aas, M. Jullum, A. Løland. – ArXiv. – URL: https://arxiv.org/pdf/1903.10464.pdf (дата обращения: 15.04.2023). – Текст: электронный.

23. Чашкин, Ю.Р. Математическая статистика. Анализ и обработка данных: Учебное пособие / Ю.Р. Чашкин. – Д: Феникс, 2017. – 236 с. – Текст непосредственный.

24. Deisenroth, M. P. Mathematics for machine learning / M. P. Deisenroth, A. A. Faisal, S. C. Ong. – Cambridge University Press, 2020. – 395 с. – Текст: непосредственный.

25. Библиотека для построения нейросетевых моделей TensorFlow: An open-source machine learning framework for everyone. – URL: https://www.tensorflow.org (дата обращения: 10.02.2023)

26. Python documentation – URL: https://docs.python.org/ (дата обращения: 10.02.2023)

Реализация модели на языке программирования Python

```
from tensorflow import keras
from keras.layers import Dense, Input, LSTM, Concatenate, Dropout,
Bidirectional
from keras.activations import elu

input_layer = Input(
    shape=(1, 37),
    name="input_layer"
)

hidden_layer = Bidirectional(
    LSTM(
        units = 10,
        name="hidden_layer",
        return_sequences = False,
        kernel_initializer = 'normal',
        activation = 'tanh'
    )
)(input_layer)

dropout = Dropout(0.3)(hidden_layer)

hidden_mu = Dense(
    units = 35,
    activation = 'tanh',
    name = "hidden_mu"
)(dropout)

hidden_sigma = Dense(
    units = 35,
    activation = 'sigmoid',
    name = "hidden_sigma"
)(dropout)

mu = Dense(
    units = 1,
    activation = "linear",
    name = "mu"
)(hidden_mu)

sigma = Dense(
    units = 1,
    activation = lambda x: elu(x) + 1,
    name = "sigma"
)(hidden_sigma)

output_layer = Concatenate(name='output_layer')([mu, sigma])

model = Model(
    inputs = input_layer,
    output = output_layer,
    name = 'bilstm_model'
)
```