

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет имени Н.Э.
Баумана
(национальный исследовательский университет)»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
по курсу
«Data Science»
Повышение точности 3D печати. Компенсация технологических
деформаций.

Слушатель

Белоусов Артем Сергеевич

Москва, 2022

Содержание

1. Введение	3
2. Постановка задачи	7
3. Описание используемых методов	10
4. Подготовка и обработка данных	13
5. Архитектура нейронной сети	20
6. Тестирование модели	22
Список используемой литературы	29
Приложение 1. Разработка приложения	30

1. Введение

Аддитивная технология (АТ) изготовления – это процесс послойного создания деталей путем добавления материала. В ряде случаев, АТ сокращают себестоимость деталей, увеличивают коэффициент использования материала, ускоряют вывод опытного или серийного изделия на рынок. Некоторые типы АТ позволяют изготавливать принципиально новые и сложные детали. На данный момент существует более 30-ти АТ, которые отличаются способом нанесения и соединения слоев, используемым материалом. При изготовлении сверхсложных металлических деталей используется технология селективного лазерного сплавления (СЛС, или англ. SLM/DMLM/LBM). Технология СЛС основана на выборочном сплавлении металлического порошка лазером в заранее сформированном слое, более подробно данная технология описана в [5].

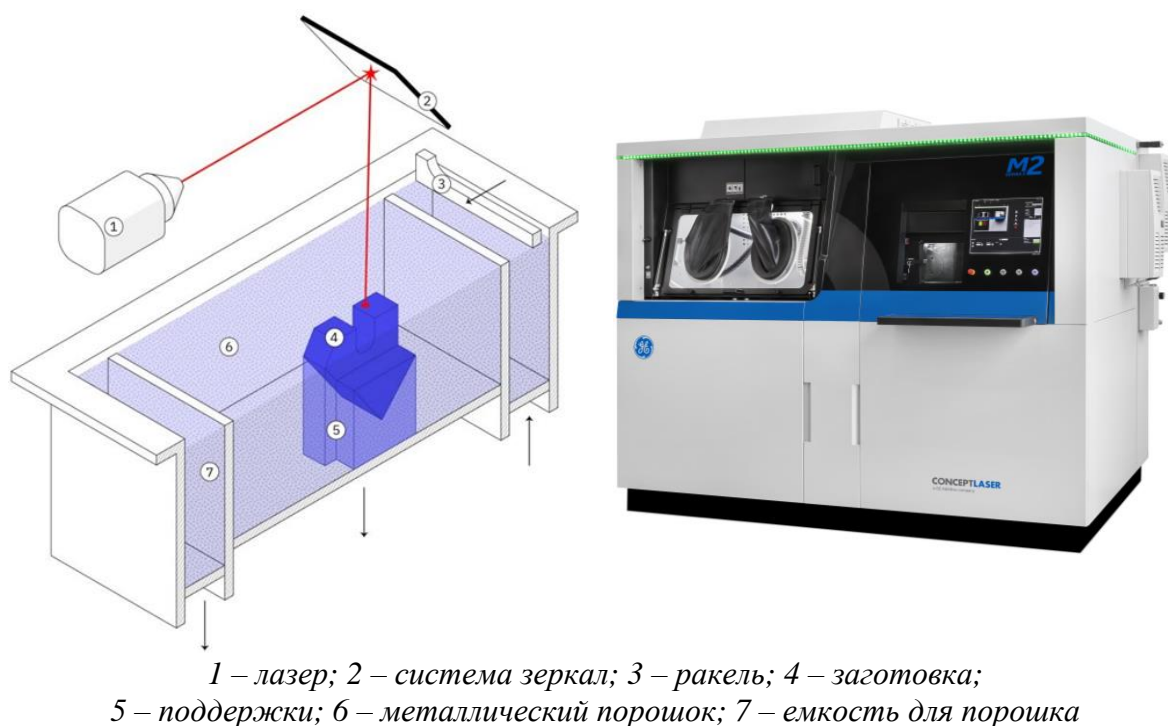


Рисунок 1.1 – Схематичный процесс СЛС и внешний вид установки СЛС.

Высокие технологические деформации или коробления – неизбежный тип дефекта, который возникает при металлической 3D-печати. С увеличением габаритов заготовки данный дефект становится все более существенным, а также снижает эффективность аддитивного производства.

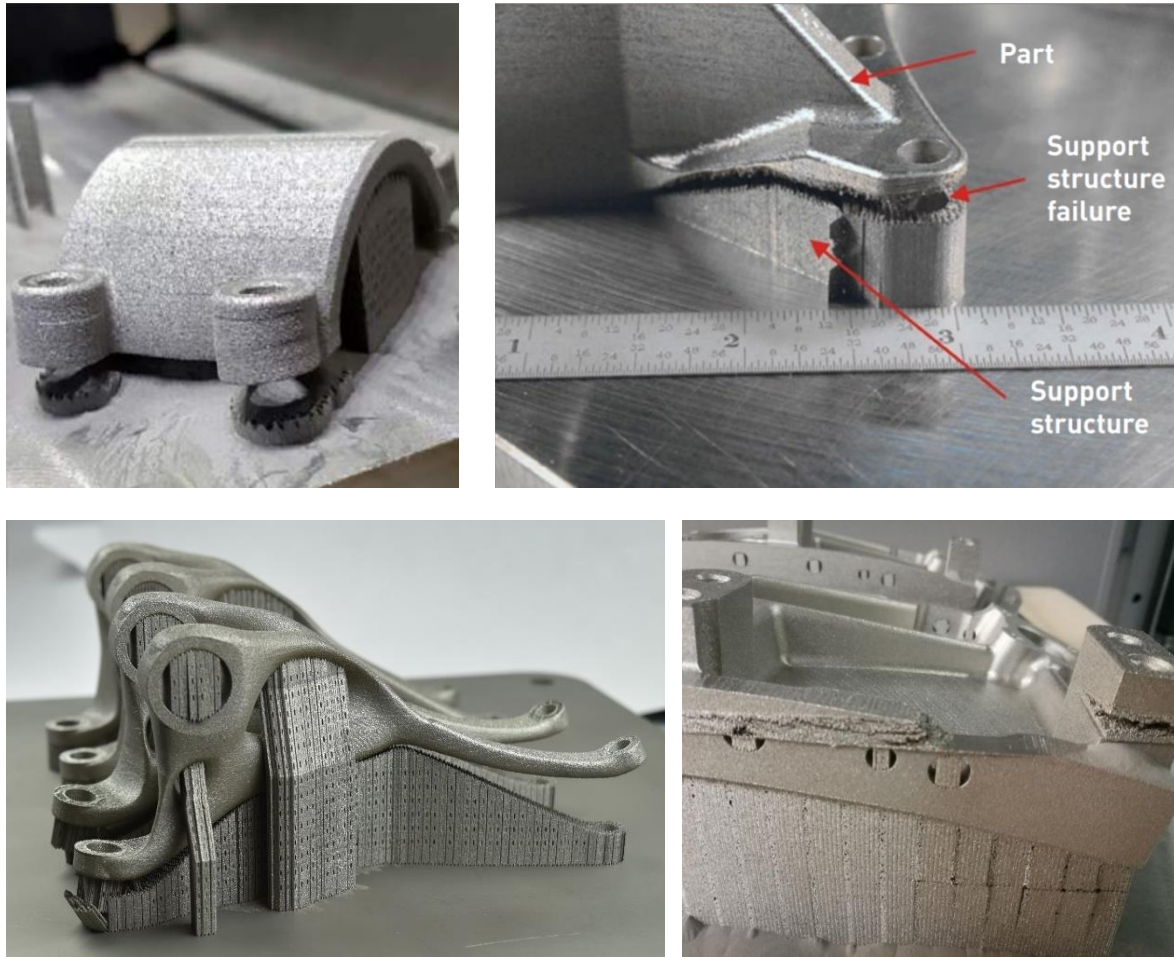


Рисунок 1.2 – Дефекты при изготовлении по технологии СЛС

Причина коробления заготовок в металлической 3D-печати связана с накоплением сжимающих деформаций, приобретенных в процессе сплавления и охлаждения множества слоев толщиной от 20 до 1000 микрон (в зависимости от технологии). Данный процесс неизбежен, также, как и технологические деформации заготовок. С ростом габаритов заготовки, сложности ее геометрии, предугадать интенсивность короблений, а также предотвратить их становится практически невозможно. Наиболее распространенное решение данной проблемы — это предварительная деформация (предеформация) заготовки перед изготовлением. Зная поведение деформаций заготовки, можно инвертировать отклонения таким образом, чтобы заготовка в конечном итоге соответствовала исходной 3D модели. Деформированное состояние заготовки возможно спрогнозировать с помощью компьютерного моделирования, а также получить методом оптического сканирования.

Процесс компьютерного моделирования достаточно трудоемкий, требует специализированного дорогостоящего ПО и квалифицированных инженеров-расчетчиков. Для получения корректного и максимально точного результата требуется широкий спектр термомеханических свойств материала вплоть до температуры плавления, получение которых отдельная трудозатратная задача. Поэтому предеформация по моделированию не всегда применима на практике, так как на реальных задачах сложно, а иногда и невозможно достичь удовлетворительного результата и получить годную деталь в допуске отклонений.

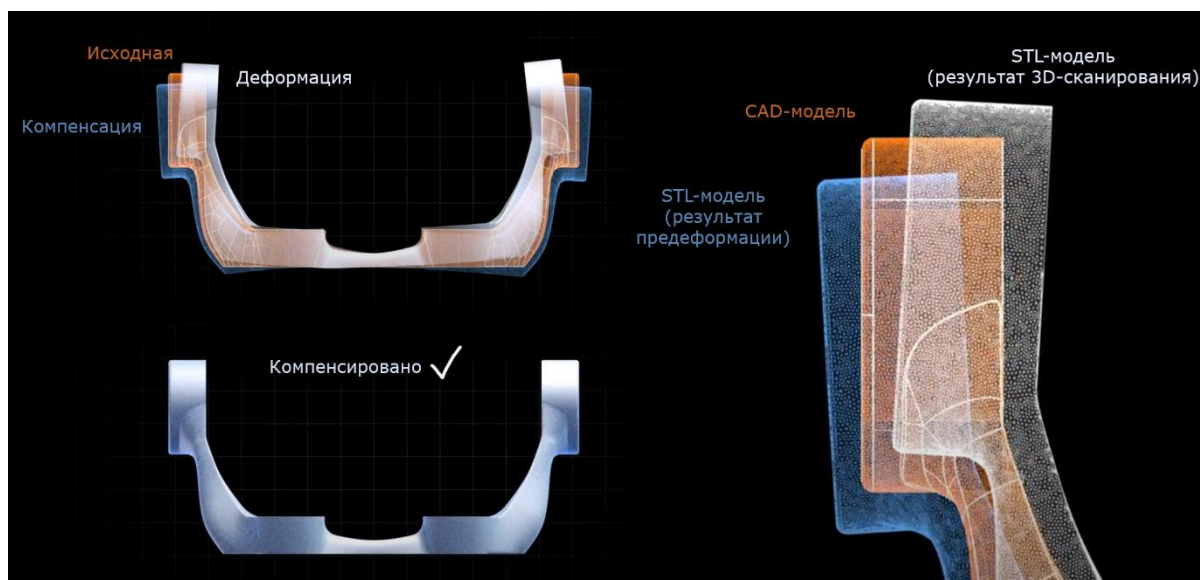


Рисунок 1.3 – Концепция предварительной компенсации технологических деформаций (предеформации)

Для предеформации по результатам 3D сканирования, автором была разработана методика, которую кратко можно описать следующим образом: после изготовления заготовка сканируется оптическим методом или компьютерной томографией (получаем STL-модель, облако точек), определяется карта отклонений, а затем отклонения инвертируются и прикладываются к исходной 3D-модели. В итоге получаем скомпенсированную поверхностную модель в формате STL.

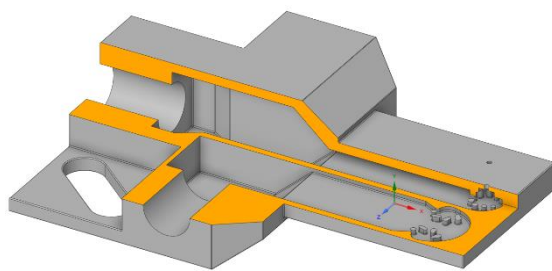
В таком подходе есть несколько сложностей. Как правило, STL модель после сканирования плохого качества: имеет множество отверстий, пересечений, неровность поверхности и т.п. Также в деталях сложной формы не представляется возможным получить данные отклонений с внутренних поверхностей и полостей при оптическом сканировании.

Поэтому требуется особая обработка и интерпретация данных, полученных после 3D сканера.

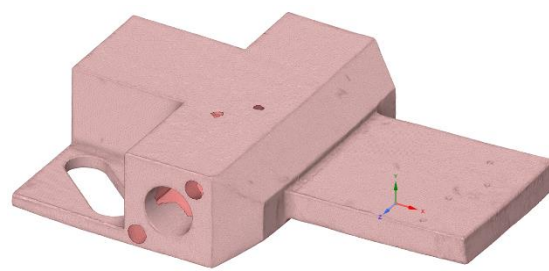
В данной работе рассмотрена реализация описанного выше подхода с помощью машинного обучения на языке программирования Python.

2. Постановка задачи

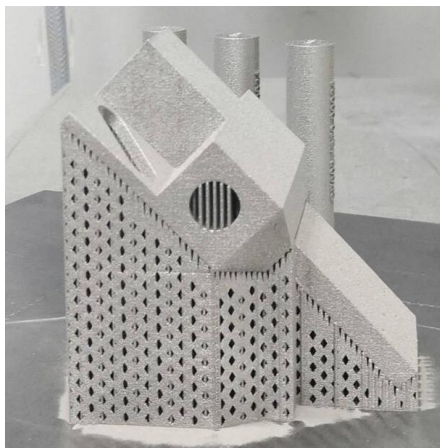
На рисунке 2.1а представлена трехмерная геометрическая модель корпуса датчика со сложной системой внутренних каналов. Заготовка изготовлена из жаропрочного никелевого сплава по технологии SLM (рис. 2.1в). После удаления и зачистки поддерживающих структур заготовка прошла контроль геометрии методом оптического 3D сканирования (рис. 2.1б) на высокоточном оборудовании GOM Atos 5, вследствие чего выявлены высокие коробления (рис. 2.1г).



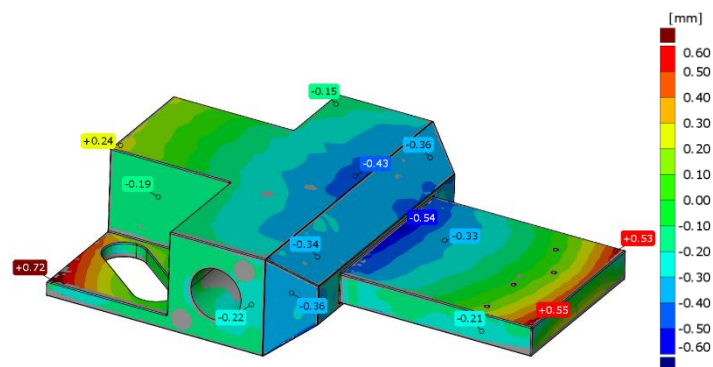
а) CAD-модель в разрезе



б) 3D-скан (STL)



в) заготовка на плите
построения с поддержками



г) коробления заготовки
(карта отклонений)

Рисунок 2.1 – Изготовление заготовки корпуса

В результате оптического сканирования создается облако точек или поверхностная модель в формате STL (*stereolithography*), которая сравнивается с исходной CAD-моделью, после чего определяется отклонение в каждой дискретной точке модели. (рис. 2.2). Из-за особенности технологии оптического сканирования невозможно распознать внутренние полости и каналы. При этом STL-моделям свойственны дефекты в виде отсутствующих граней, несплошностей,

пересечений и т.д. К тому же, на практике заготовки деталей имеют локальные неровности, шероховатость поверхности, остатки от поддерживающих структур, что в свою очередь отражается на результатах измерений.

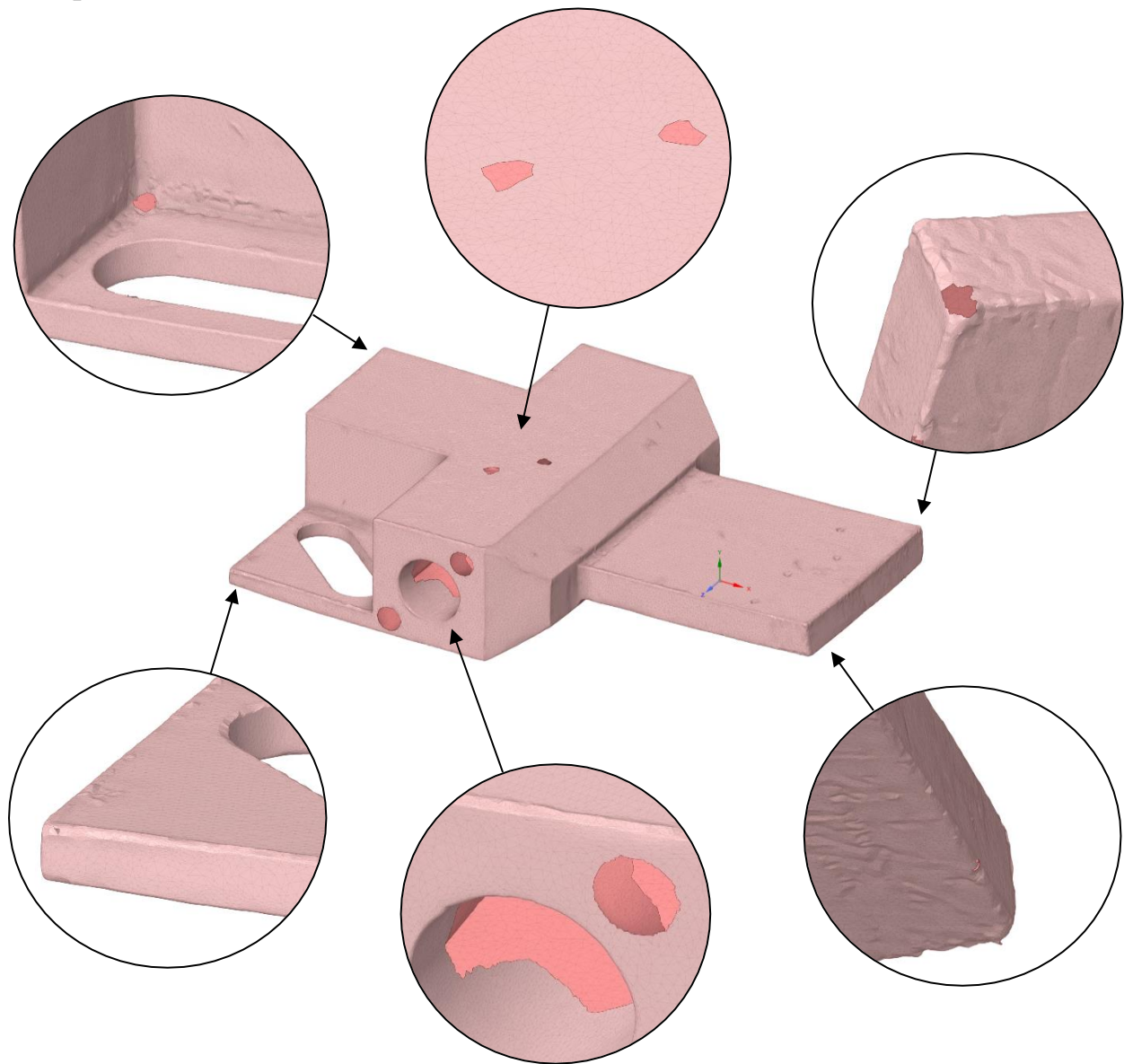


Рисунок 2.2 – Типичные дефекты STL-модели, полученной в результате 3D-сканирования

Исходя из данных 3D сканирования, необходимо выполнить предварительную деформацию исходной геометрической модели с инверсией короблений для последующего изготовления и получения заготовки с допустимыми отклонениями.

При реализации методики компенсации технологических короблений необходимо:

- определять характер деформаций на качественном и количественном уровне
- предсказывать отклонения в точках, где нет данных измерений (внутренние полости, дефекты STL-модели)
- не переносить на предеформированную модель локальные дефекты (шероховатость, неровности, остатки от поддержек и др.)

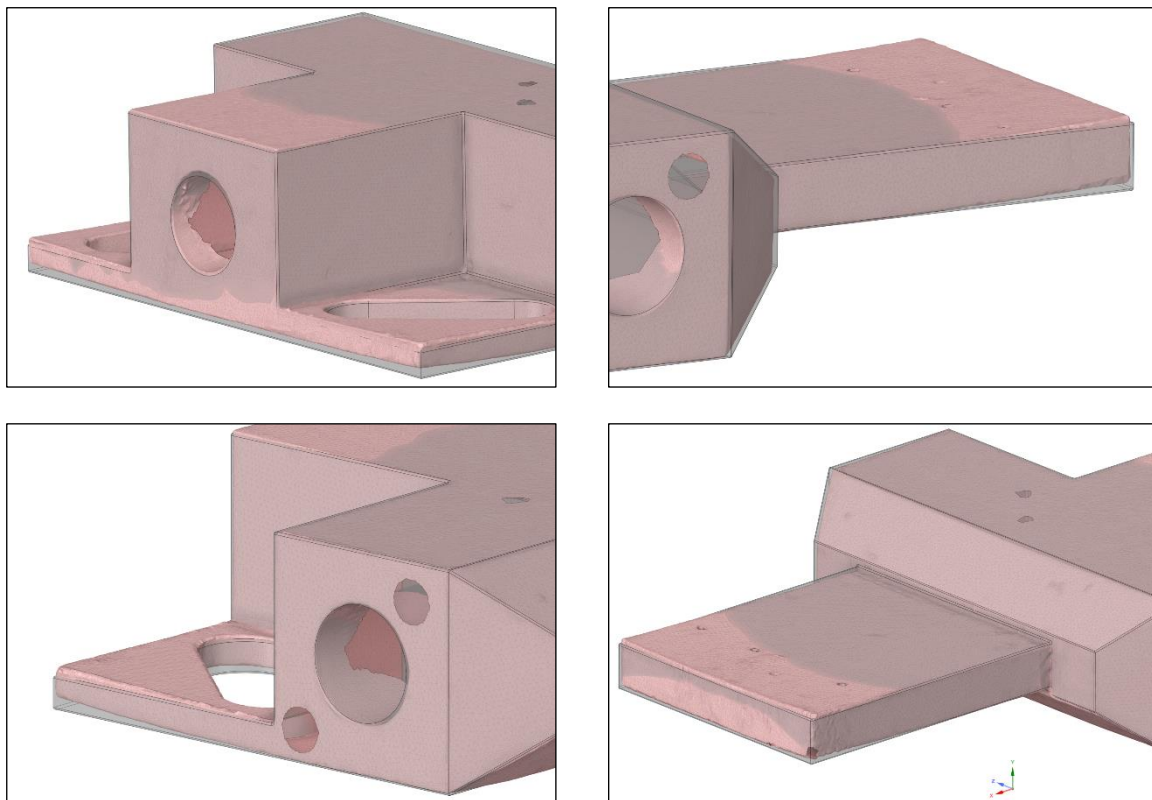


Рисунок 2.3 – Сравнение CAD-модели и STL-скана

В качестве исходных данных выступают 3D модель заготовки и файл с результатами отклонений (в формате **csv**).

3. Описание используемых методов

Нормализация данных в данной работе осуществлялась с помощью метода *MinMaxScaler* из библиотеки ScikitLearn. Суть заключается в том, что метод масштабирует и преобразует каждую переменную по отдельности, чтобы она находилась в заданном диапазоне на обучающем наборе, например, между нулем и единицей. Преобразования осуществляются по следующему выражению:

$$X_{nrm} = \frac{X - X_{min}}{X_{max} - X_{min}} \times (f_{max} - f_{min}) + f_{min}, \quad (3.1)$$

где X_{nrm} – нормализованное значение; X – значение переменной датасета; X_{min} – минимальное значение переменной; X_{max} – максимальное значение переменной; $f = (f_{min}, f_{max})$ – заданный диапазон для нормализации.

Масштабирование признаков осуществлялось методом *QuantileTransformer* из библиотеки ScikitLearn. Этот метод преобразует объекты так, чтобы они соответствовали равномерному или нормальному распределению. Преобразование применяется к каждому объекту независимо. Сначала используется оценка кумулятивной функции распределения признака, чтобы сопоставить исходные значения с однородным распределением. Затем полученные значения сопоставляются с желаемым выходным распределением с помощью соответствующей функции квантиля. Значения признаков новых/непросмотренных данных, которые находятся ниже или выше установленного диапазона, будут сопоставлены с границами выходного распределения. Стоит учитывать, что это преобразование является нелинейным. Оно может исказить линейные корреляции между переменными, измеренными в одном масштабе, но делает переменные, измеренные в разных масштабах, более сопоставимыми.

Выбросы исключались по правилу трех сигм. Правило трех сигм заключается в том, что при нормальном распределении практически все значения величины с вероятностью 99,73% лежат не далее трех сигм (рис.3.1) в любую сторону от математического ожидания, то есть находятся в диапазоне:

$$[\mu - 3\sigma; \mu + 3\sigma], \quad (3.2)$$

где μ – математическое ожидание; σ – стандартное отклонение.

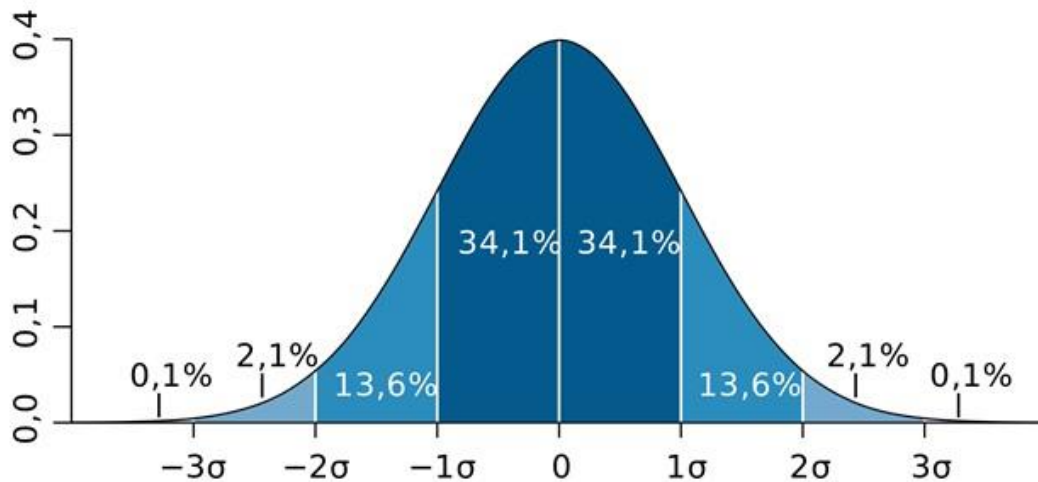


Рисунок 3.1 – График нормального распределения

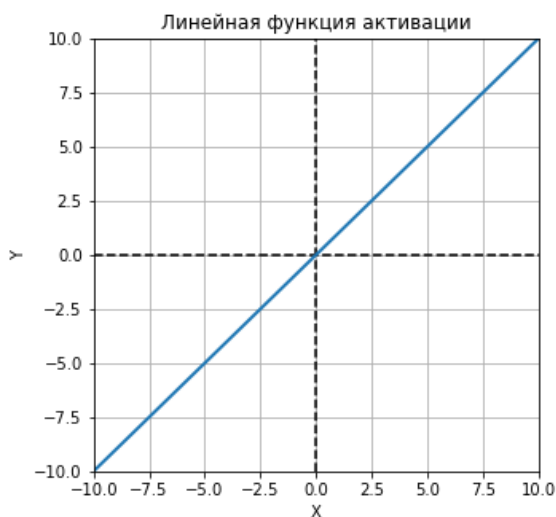
Для решения задачи регрессии была использована многослойная полносвязная искусственная нейронная сеть (ИНС) по принципу обучения с учителем (рис.3.3). Использовались два вида функций активаций:

- линейная для входного и выходного слоя нейронов (рис.3.2а):

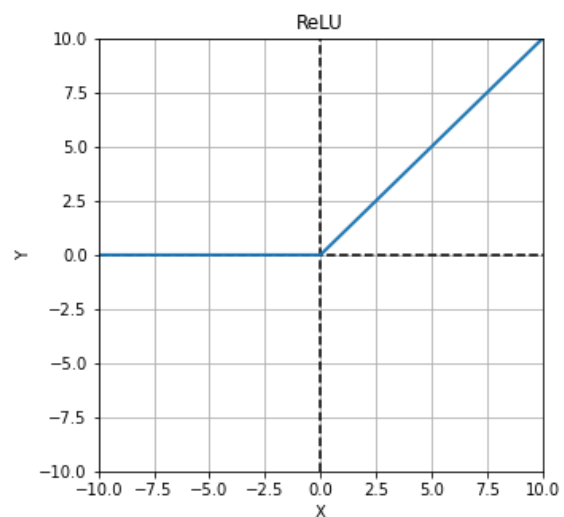
$$f(x) = x, \text{ при } x \in (-\infty, +\infty), \quad (3.3)$$

- функция *ReLU* для скрытых (промежуточных) слоев (рис.3.2б):

$$f(x) = \begin{cases} 0 & x < 0 \\ x & x \geq 0 \end{cases}, \text{ при } x \in [0, +\infty). \quad (3.4)$$



а) линейная функция активации



б) функция активации ReLU

Рисунок 3.2 – Графики функции активации

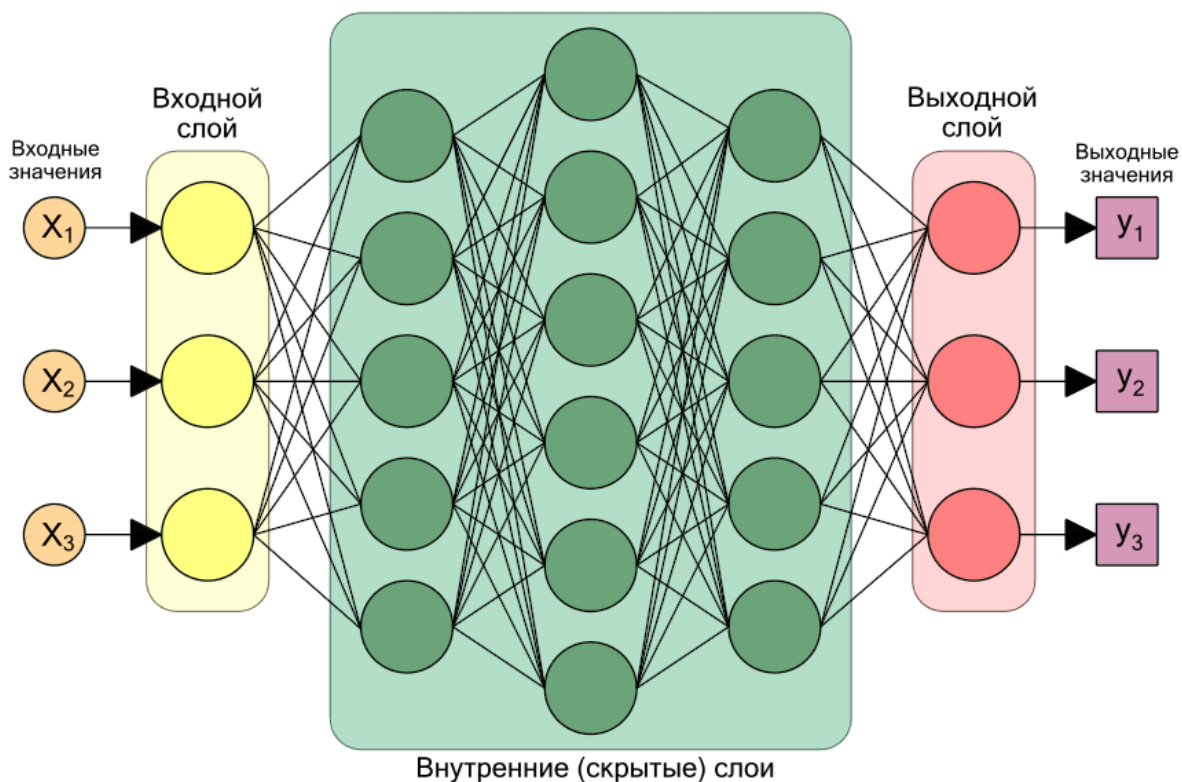


Рисунок 3.3 – Схема многослойной искусственной нейронной сети

В алгоритме оптимизации для минимизации ошибки использовались следующие метрики:

- Средняя абсолютная ошибка (*Mean Absolute Error, MAE*), которая вычисляется как:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|, \quad (3.5)$$

где y_i – истинное значение; \hat{y}_i – предсказанное значение; n – количество значений.

- Среднеквадратичная ошибка (*Mean Squared Error, MSE*), которая вычисляется как:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2, \quad (3.6)$$

где y_i – истинное значение; \hat{y}_i – предсказанное значение; n – количество значений.

4. Подготовка и обработка данных

Данные отклонений (рис. 4.1), полученные в результате 3D сканирования, содержат: три столбца с координатами точек («X», «Y», «Z») в которых осуществлялись измерения; три столбца с вектором нормали в каждой точке («nx», «ny», «nz»); три столбца с отклонениями в каждой точке по трем осям («dx», «dy», «dz»); столбец с суммарным отклонением («dev»). Тип данных во всех столбцах – число с плавающей точкой (float), а физически данные представляют измерения в миллиметрах.

	X	Y	Z	nx	ny	nz	dx	dy	dz	dev
0	-39.000000	5.623463	-3.734776	-0.011177	-0.412853	-0.910729	-0.002699	0.091168	0.203386	-0.222901
1	-39.000000	5.660982	-3.746157	-0.011951	-0.227208	-0.973773	0.003678	0.060979	0.211567	-0.220210
2	-39.000000	5.700000	-3.750000	-0.012212	-0.098010	-0.995111	0.010419	0.026891	0.214464	-0.216394
3	-38.895326	5.623463	-3.736061	-0.022531	-0.382790	-0.923561	-0.002729	0.092166	0.205612	-0.225341
4	-38.894873	5.588886	-3.717584	-0.020196	-0.555708	-0.831132	0.000833	0.107823	0.195670	-0.223413

Рисунок 4.1 – Исходные данные (первые 5 строк)

Создадим описательную статистику датасета и определим количество пропущенных значений с помощью методов *.describe()* и *.isna().sum()* из библиотеки Pandas.

	X	Y	Z	nx	ny	nz	dx	dy	dz	dev		
count	449427.000	449427.000	449427.000	449427.000	449427.000	449427.000	166009.000	166009.000	166009.000	166009.000	X	0
mean	-10.561	14.938	15.644	0.001	-0.020	-0.006	-0.006	-0.026	0.000	-0.103	Y	0
std	28.195	5.622	15.706	0.460	0.718	0.521	0.086	0.222	0.094	0.236	Z	0
min	-58.000	5.500	-15.500	-1.000	-1.000	-1.000	-0.880	-0.572	-0.658	-1.106	nx	0
25%	-32.966	10.887	4.208	-0.000	-0.800	-0.031	-0.003	-0.184	-0.002	-0.262	ny	0
50%	-14.115	15.150	15.288	0.000	0.000	0.000	0.000	-0.002	0.000	-0.123	nz	0
75%	21.188	18.500	27.342	0.000	0.741	0.000	0.008	0.019	0.002	0.063	dx	283418
max	35.000	25.500	48.000	1.000	1.000	1.000	0.987	1.069	2.549	2.553	dy	283418
											dz	283418
											dev	283418

Рисунок 4.2 – Описательная статистика исходного датасета

Общее количество строк исходного датасета равно 449 427, больше половины значений в котором имеют пропуски (248 418). После удаления строк, останется 166 009 значений, что является вполне достаточным количеством для рассмотрения и обучения НС. Визуализация пропусков осуществлена с помощью матричной диаграммы (тепловая карта, метод *heatmap()*) из библиотеки Seaborn. Как видно из диаграммы (рис. 4.3), значения отсутствуют только в столбцах с отклонениями, причем если и отсутствует, то сразу во всех четырех столбцах («dx», «dy», «dz», «dev»).

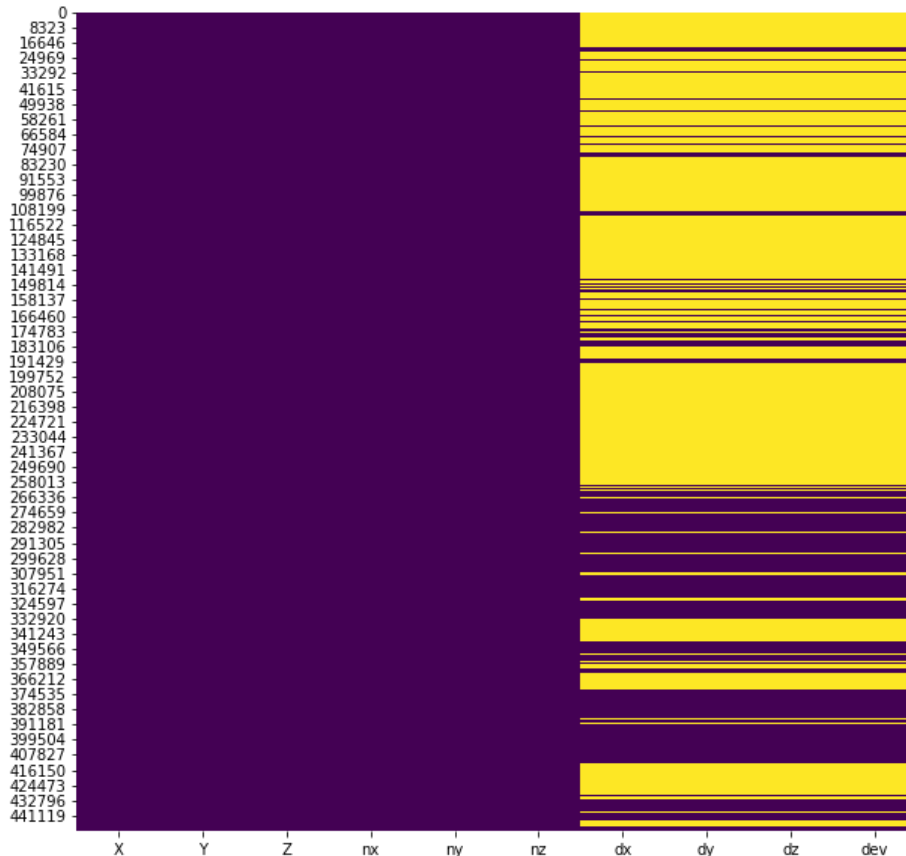


Рисунок 4.3 – Визуализация пропущенных значений
(желтый цвет – значение отсутствует)

Исходя из постановки задачи и описательной статистики, можно сделать вывод, что пропуски данных отклонений присутствуют в достаточном количестве. По условиям задачи требуется определить неизвестные отклонения, поэтому заполнять пропуски средним или медианным значением не корректно. В связи с этим допускается удаление всех строк с пропущенными значениями (метод *.dropna()*).

После удаления пропусков был проведен количественный и статистический анализ на предмет наличия выбросов, определение вероятностного распределения и зависимостей между переменными.

Для построения вероятностного распределения был использован метод *histplot()* из библиотеки Seaborn для переменных «dx», «dy», «dz». Общее отклонение «dev» не рассматривалось, так как признак не содержит информацию о направлении вектора отклонений, а также напрямую зависит от «dx», «dy», «dz» и вычисляется как:

$$dev = \sqrt{dx^2 + dy^2 + dz^2}. \quad (4.1)$$

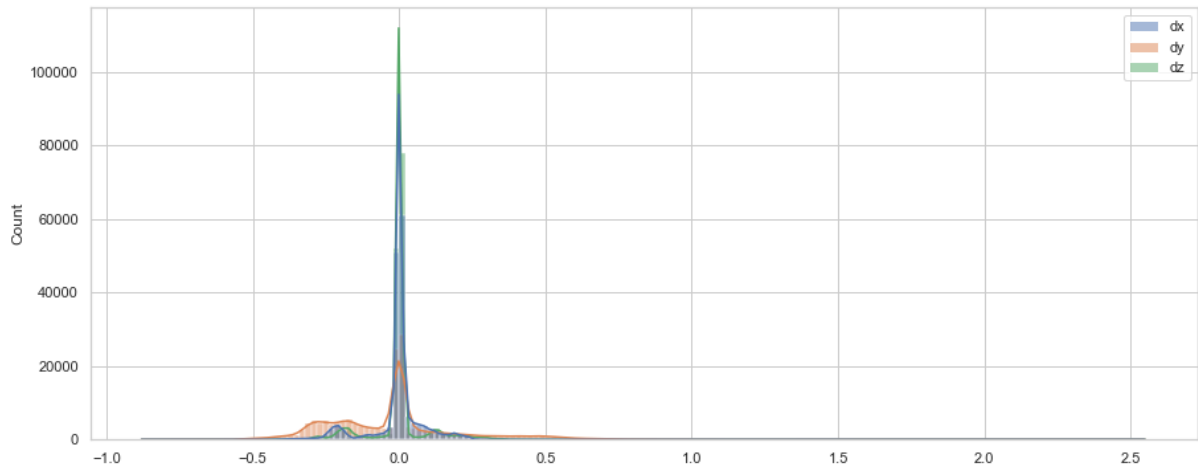


Рисунок 4.4 – Гистограмма вероятностного распределения необработанных данных

Основываясь на график, построенный на исходных необработанных данных (рис. 4.4), выявлено, что требуется масштабирование, потому как подавляющее большинство значений равны 0 или являются околонулевыми. Приведение к нормальному вероятностному распределению осуществлялось методом *QuantileTransformer* из библиотеки *ScikitLearn*, так как метод работает не только с положительными значениями, но и с отрицательными (рис. 4.5).

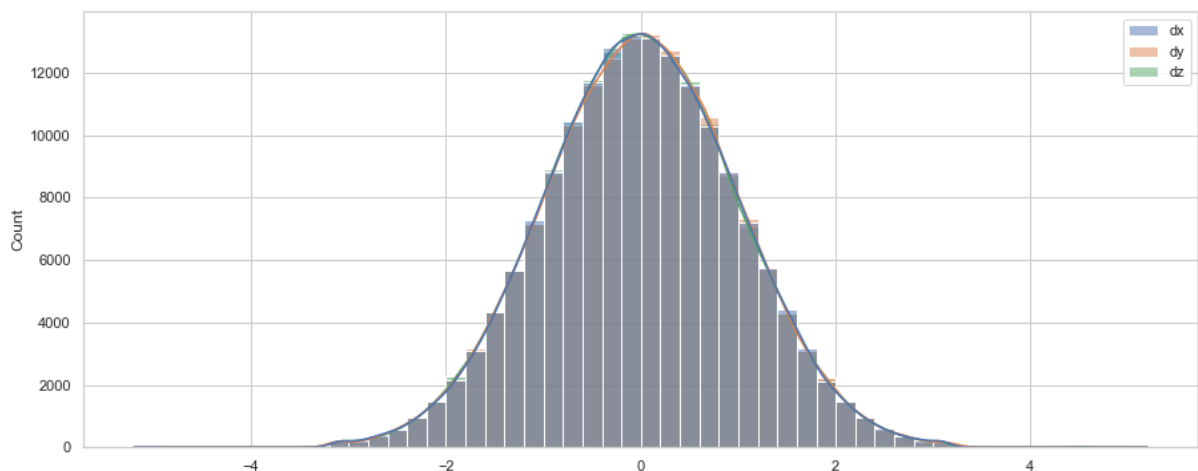


Рисунок 4.5 – Гистограмма вероятностного распределения после масштабирования

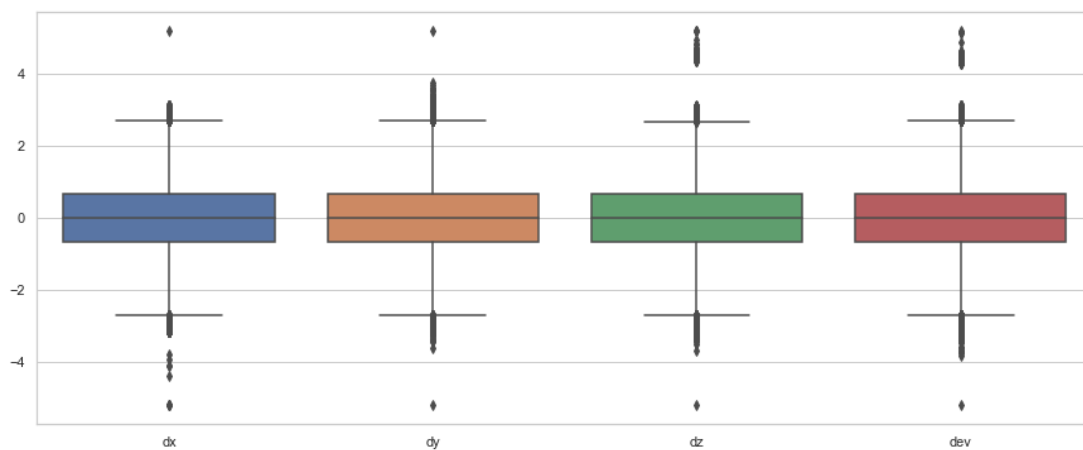


Рисунок 4.6 – Диаграмма размаха признаков масштабированных данных

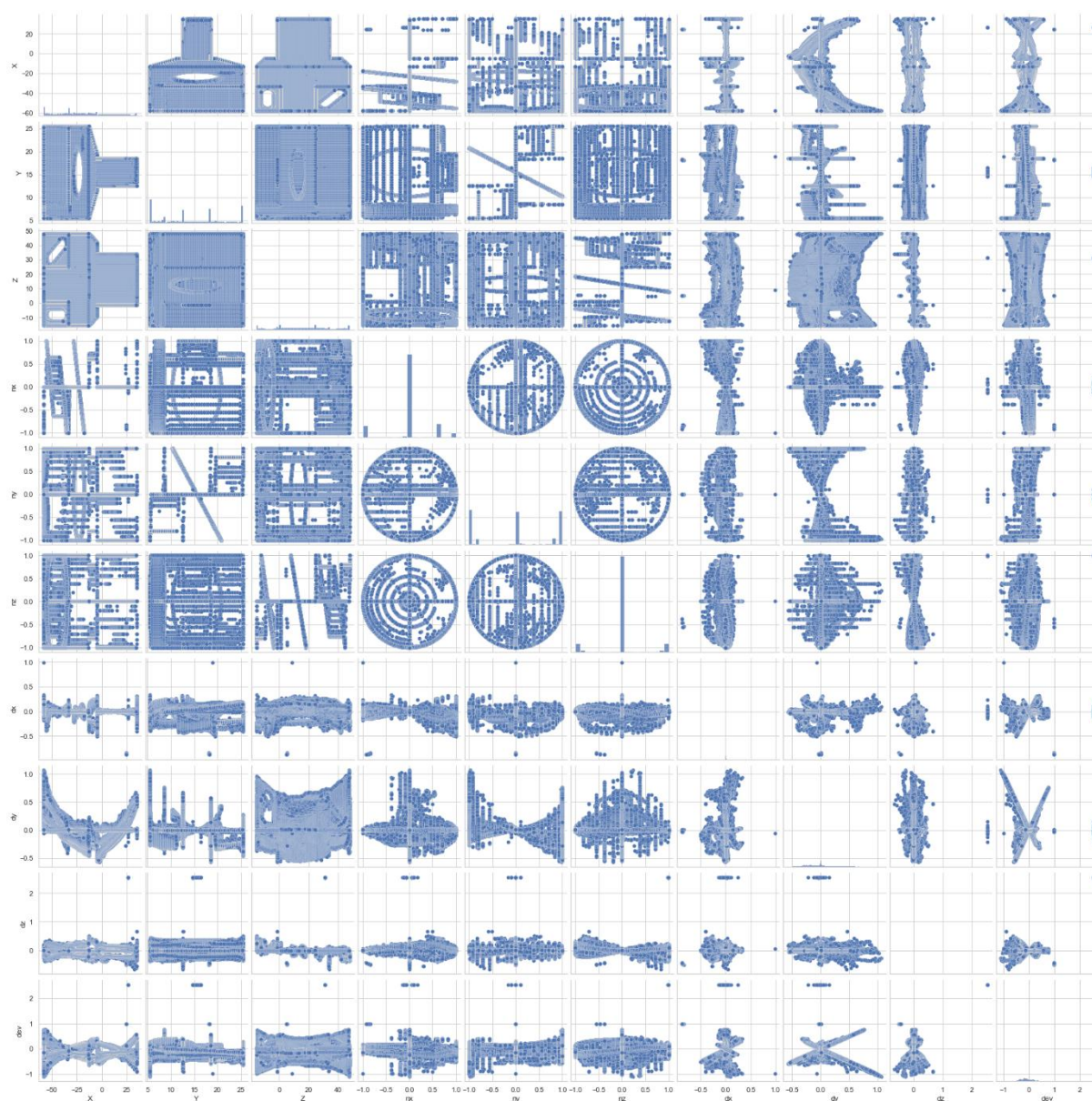
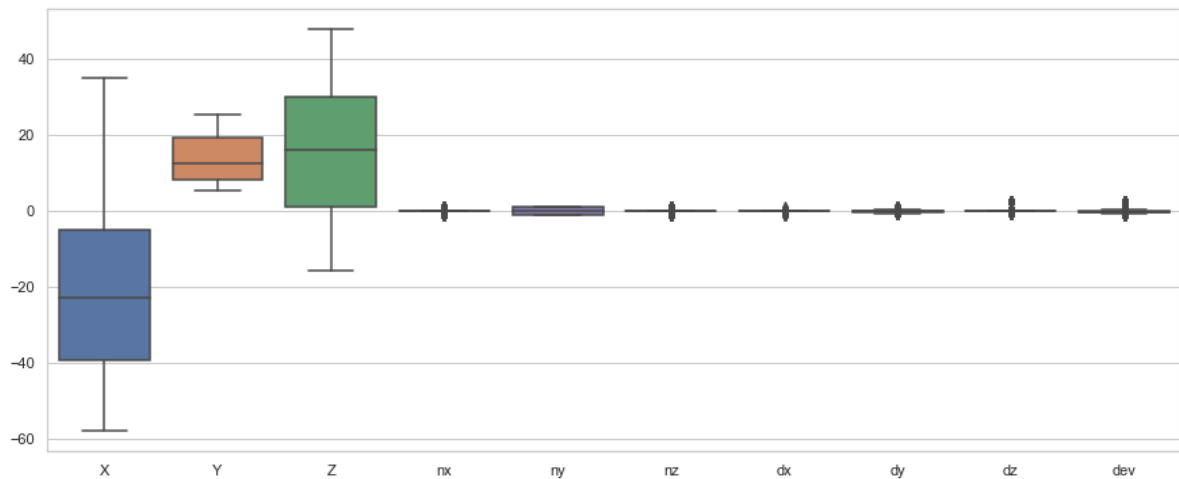
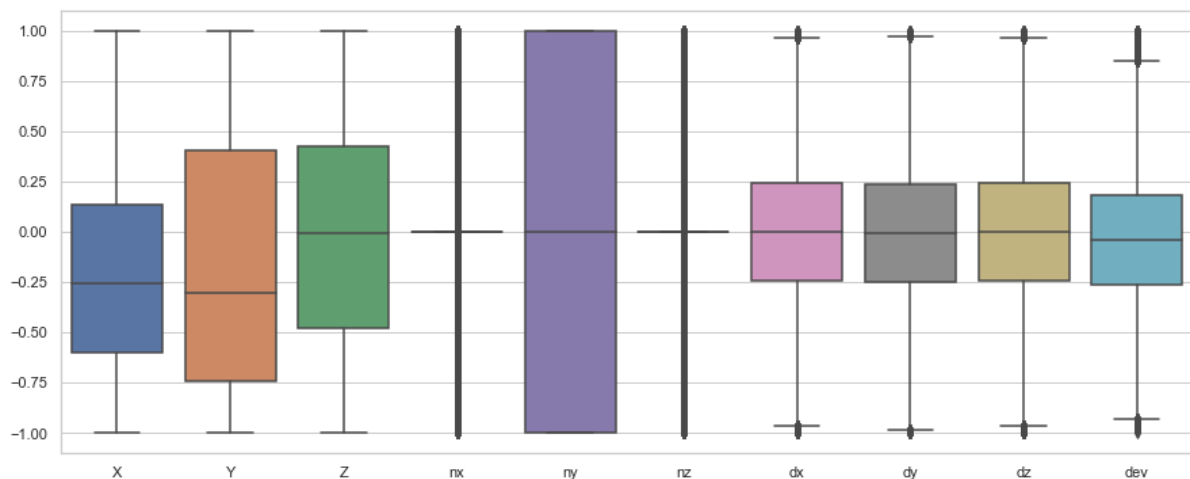


Рисунок 4.7 – График попарного распределения переменных

Диаграмма размаха (рис. 4.6) и график попарного распределения переменных (рис. 4.7), визуально показали наличие выбросов в колонках «dx», «dy», «dz», «dev». Удаление выбросов производилось по методу 3-х сигм. После чего, проведена нормализация всех переменных датасета к диапазону значений $[-1; 1]$ (рис. 4.8).



а) – исходный датасет, необработанные данные



б) – данные после масштабирования, удаления выбросов и нормализации

Рисунок 4.8 – Диаграмма размаха признаков

Проанализировав график попарного распределения переменных после масштабирования, удаления выбросов и нормализации, можно сделать следующие выводы (рис. 4.9):

- удалены явные выбросы;
- парные графики по «X», «Y», «Z» показывают виды деталей спереди, сбоку, сверху соответственно;
- зависимость «X», «Y», «Z» от «dx», «dy», «dz» отображает деформации детали по каждой из осей;
- нелинейные корреляции, наблюдаемые в «dev», на самом деле линейные, так как использовался нелинейный метод масштабирования данных *QuantileTransformer*.

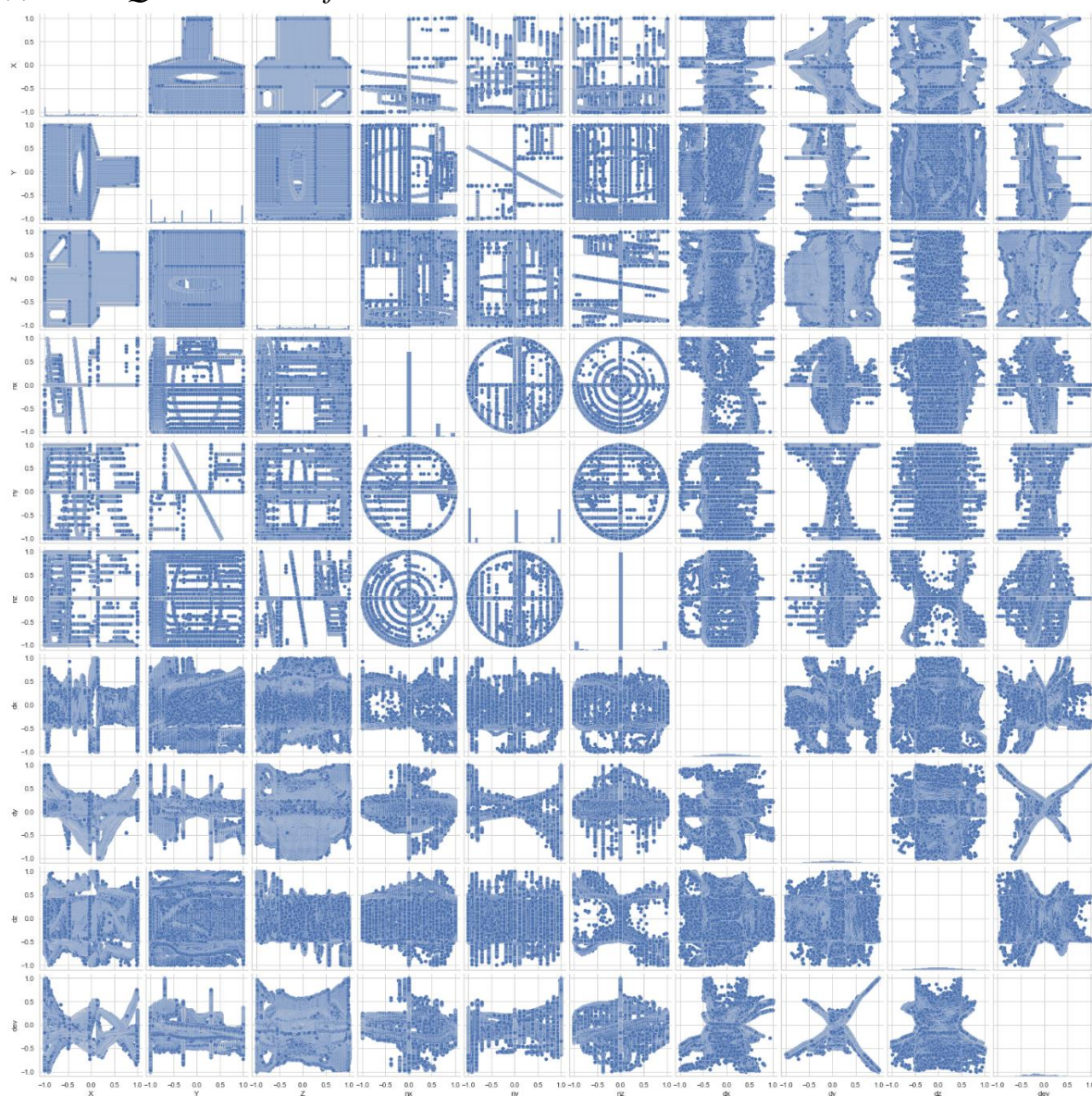


Рисунок 4.9 – График попарного распределения переменных после масштабирования, удаления выбросов и нормализации

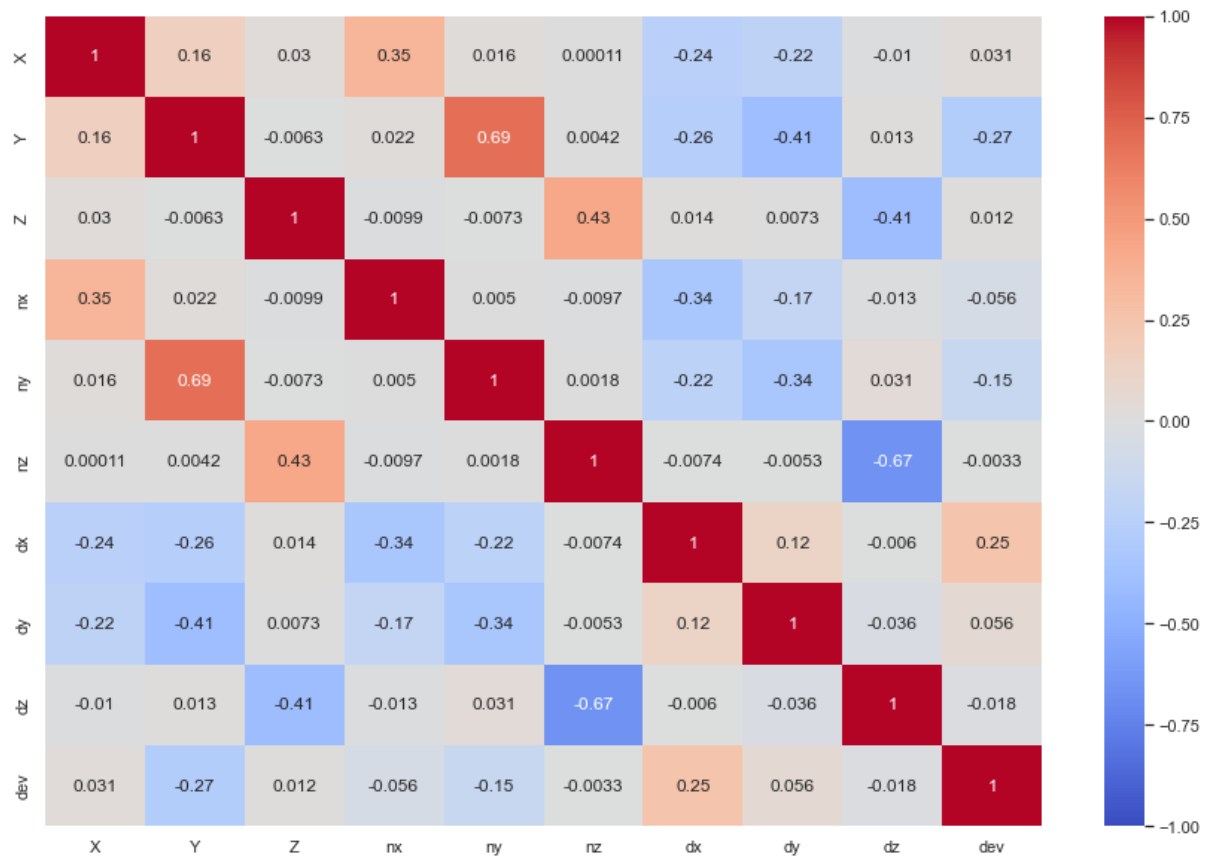


Рисунок 4.10 – Цветовая карта корреляций переменных

Анализируя карту корреляций переменных, представленной на рисунке 4.10, логично выявляются корреляции между признаками, связанными одной осью координат. Например, есть явная корреляция между признаками «X», «nx» и «dx».

5. Архитектура нейронной сети

Для прогнозирования отклонений была выбрана модель полносвязной нейронной сети, так как поставленная задача, это задача регрессии. Обучение нейронной сети выполнялось на размеченных, масштабированных и нормализованных данных с 6-ю переменными. Где на вход подавались координаты точки по осям X, Y, Z, а на выходе отклонение в этой точке аналогично по трем осям. Другими словами, мы обучаем алгоритм какими должны быть отклонения в каждой точке пространства, чтобы НС предсказывала отклонения в тех точках 3D-модели, где нет данных сканирования. Датасет делился на обучающую и тестовую выборку в соотношении 85 к 15%. Размерность выборок составляла (138617, 3) и (24462, 3) соответственно.

	X	Y	Z	dx	dy	dz
0	-0.591398	-0.987654	-0.629442	-0.214771	0.317822	0.669234
1	-0.591398	-0.983902	-0.629800	0.177710	0.289072	0.681167
2	-0.591398	-0.980000	-0.629921	0.266282	0.252713	0.685651
3	-0.589147	-0.987654	-0.629482	-0.216302	0.318711	0.672326
4	-0.589137	-0.991111	-0.628900	0.051138	0.331945	0.656628

Входные данные

Выходные данные

Рисунок 5.1 – Данные для обучения нейронной сети

Стартовая архитектура нейронной сети состояла из трех слоев: входной слой из 3 нейронов с линейной функцией активации; выходной слой из 3 нейронов в линейной функцией активации; и один скрытый слой с 16 нейронами с функцией активации «RELU» (рис. 5.2).

Layer (type)	Output Shape	Param #
=====		
dense_10 (Dense)	(None, 3)	12
dense_11 (Dense)	(None, 16)	64
dense_12 (Dense)	(None, 3)	51
=====		
Total params: 127		
Trainable params: 127		
Non-trainable params: 0		

Рисунок 5.2 – Стартовая архитектура нейронной сети

Выбран алгоритм оптимизации «Adam», как наиболее распространенный. Метрика для минимизации ошибки выбрана MSE (среднеквадратичная ошибка), и дополнительно отслеживалась метрика MAE (средняя абсолютная ошибка). Критерием для остановки алгоритма является количество эпох (3 эпохи) без улучшения выбранной метрики на обучающей выборке.

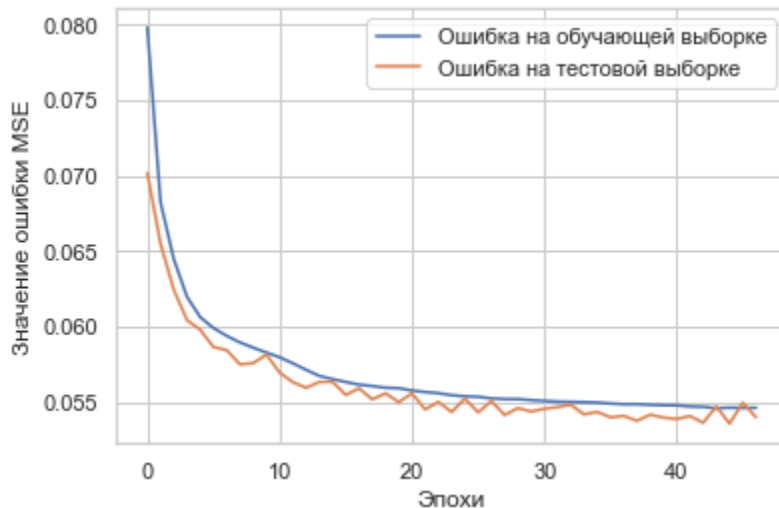


Рисунок 5.3 – Процесс обучения нейронной сети

Стартовое обучение завершилось за 47 итераций (эпох). После чего для предсказания через метод *predict()* подавались нормализованные координаты точек, для которых требуется определить отклонения. Предсказанные значения прошли процедуру обратной нормализации и обратного масштабирования чтобы получить конкретные целевые значения отклонений. Затем отклонения вычитались из значений координат исходной 3D-модели, чтобы получить модель с инверсией отклонений (предеформированную модель).

6. Тестирование модели

Проведено исследование как влияют гиперпараметры нейронной сети на конечный результат предеформирования. Было исследовано влияние количества слоев, количество нейронов, наличие слоев *BatchNormalization* и *Dropout* на качество 3D-модели (рис. 6.1-6.3).

При увеличении количества скрытых слоев нейронной сети качественно и количественно улучшается предеформация 3D-модели. Увеличение слоев свыше 3, не наблюдается улучшения в качестве. С увеличением количества нейронов на каждом слое также прослеживается повышается точность прогнозирования. Но увеличение свыше 256 нейронов, приводит к эффекту переобучения.

С добавлением слоя *BatchNormalization*, наблюдается общее улучшение качества поверхности геометрической модели, но количественно точность ухудшается. А с добавлением дополнительного слоя *Dropout* количественно достигается требуемая точность, причем значительно увеличивается качество прогнозирования во внутренних каналах.

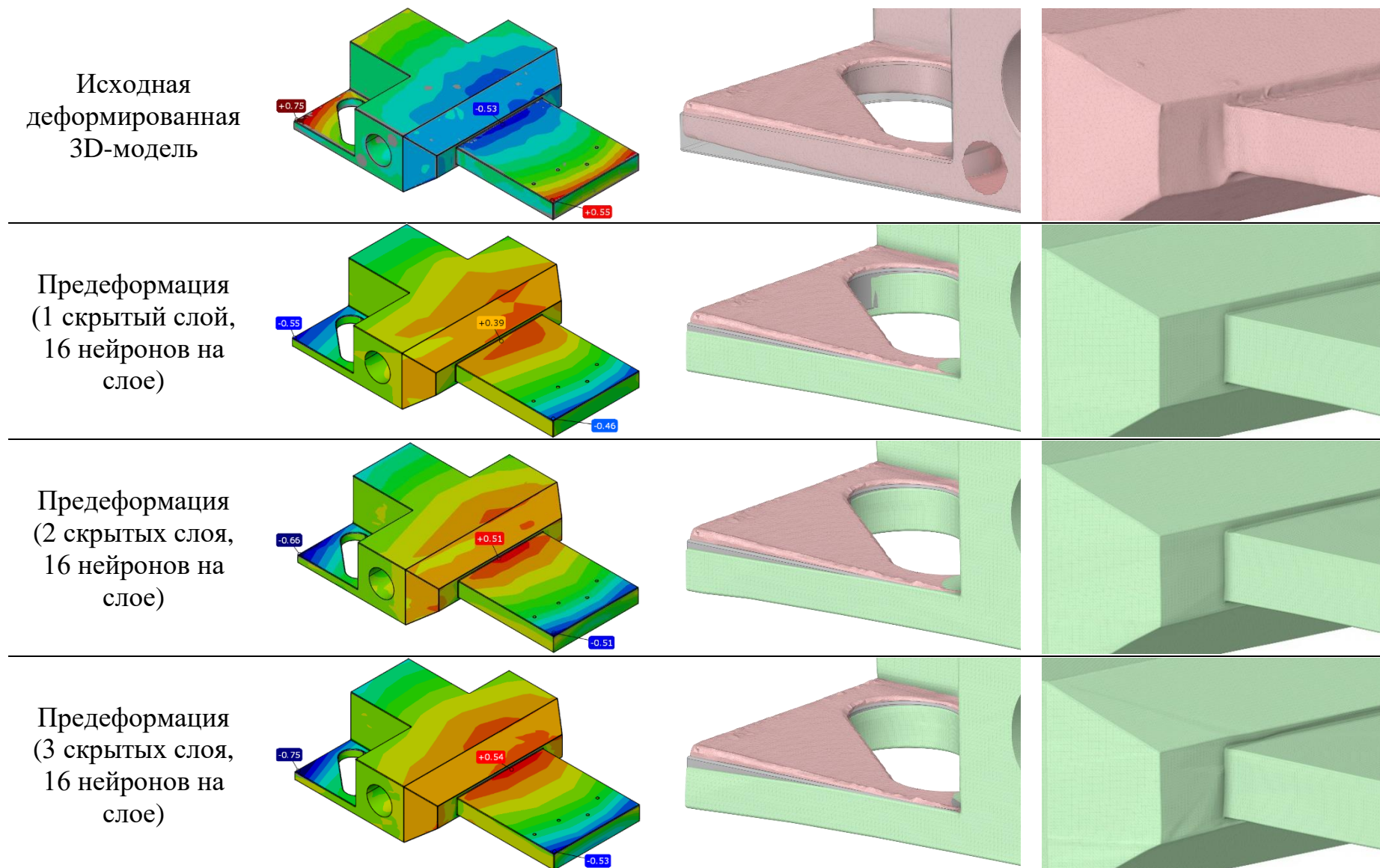


Рисунок 6.1 – Исследование влияния количества скрытых слоев нейронной сети на качество 3D-модели

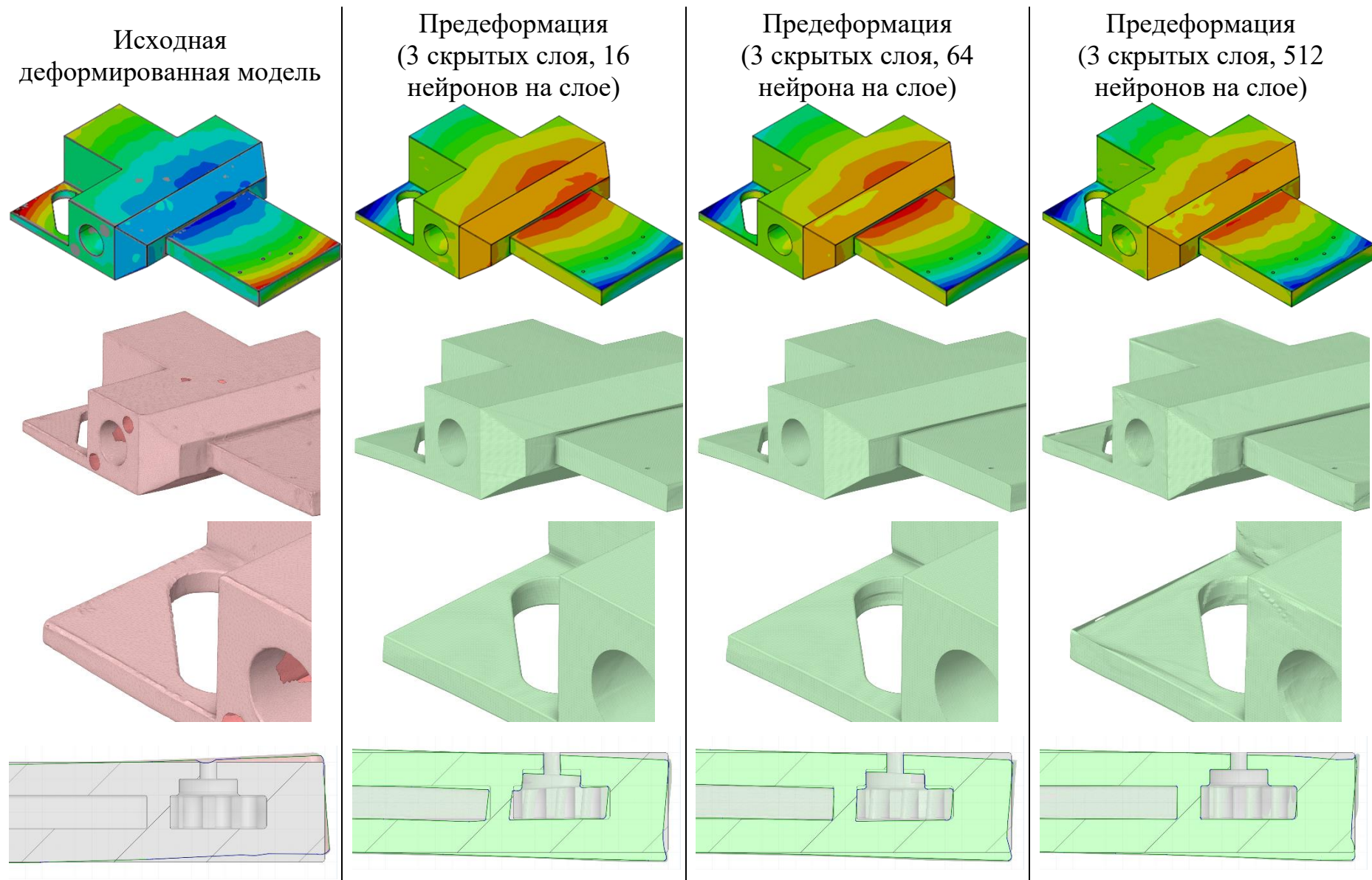
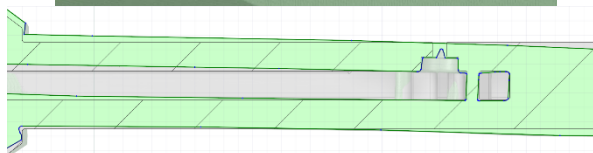
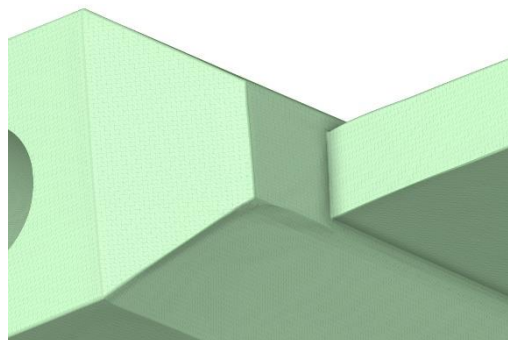
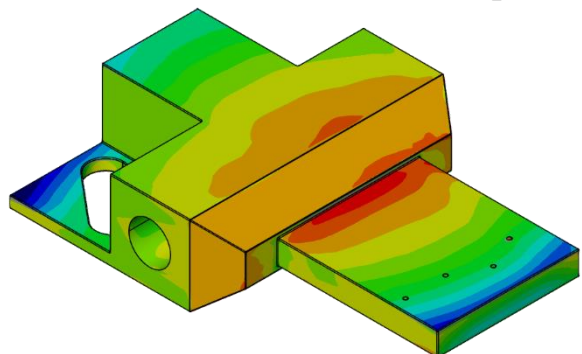
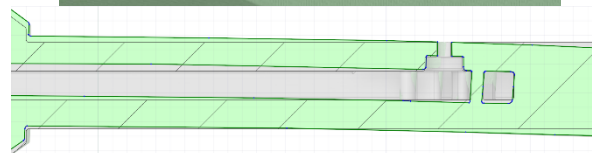
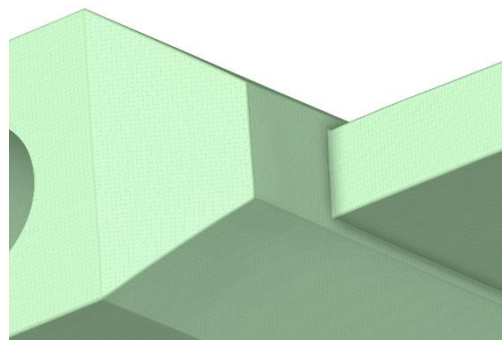
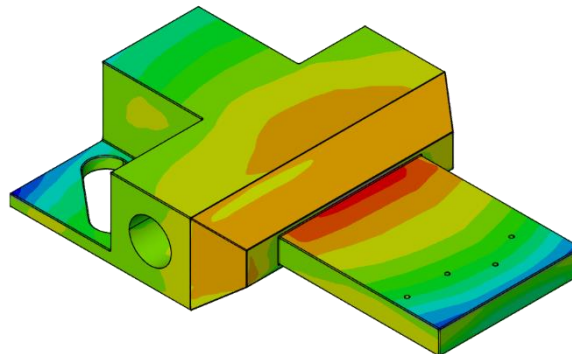


Рисунок 6.2 – Исследование влияния количества нейронов на качество 3D-модели

Без BatchNormalization и Dropout



BatchNormalization



BatchNormalization и Dropout

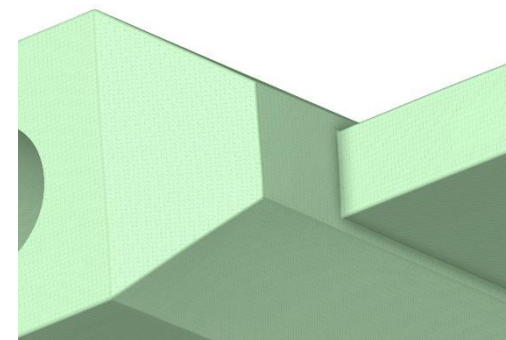
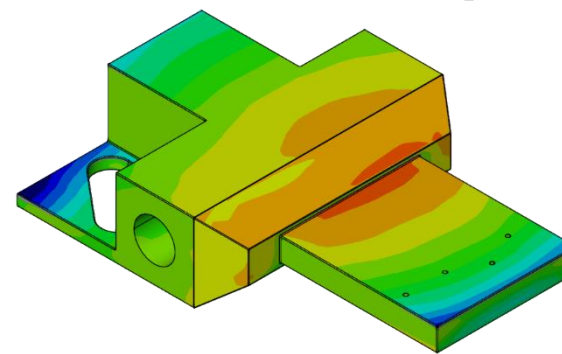


Рисунок 6.3 – Исследование влияния наличия BatchNormalization и Dropout слоев на качество 3D-модели

В конечном итоге, была выбрана архитектура нейронной сети с 3 скрытыми слоями по 64 нейрона на каждом, где после входного слоя добавлен слой BatchNormalization, а перед выходным слоем – слой Dropout. Создана предеформированная 3D-модель, по которой была изготовлена новая деталь по технологии SLM, по тому же технологическому процессу, что и исходная деталь. Максимальные коробления снизились с 0.7 до 0.1 мм (рис. 6.4).

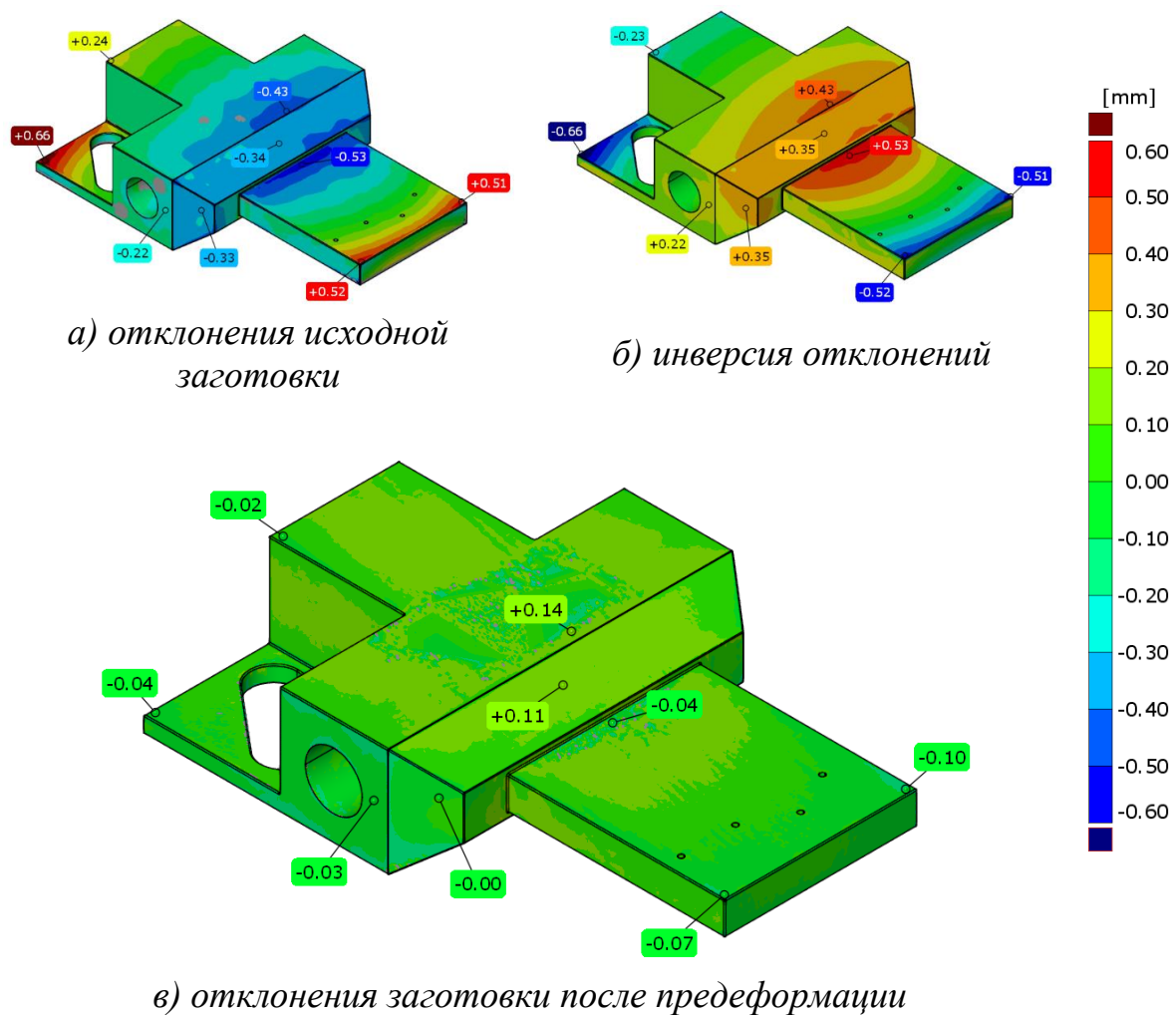


Рисунок 6.4 – Результат изготовления предеформированной модели

Реализованная методика дополнительно была протестирована на детали «корпус» (рис. 6.5). Начальные деформации составляли около 0.6 мм, после чего была создана и изготовлена предеформированная модель, коробления которой удалось снизить до 0.13 мм.

Также выполнена предеформация детали «лопатка» со сложнопрофильными поверхностями (рис. 6.6). Реализованный алгоритм предеформации на основе машинного обучения позволяет качественно определять характер деформаций, при этом игнорируя локальные дефекты и неровности поверхности. За счет применения предеформации возможно в несколько раз повысить точность изготовления, соответственно существенно снизить экономические затраты на дефектное изготовление и брак.

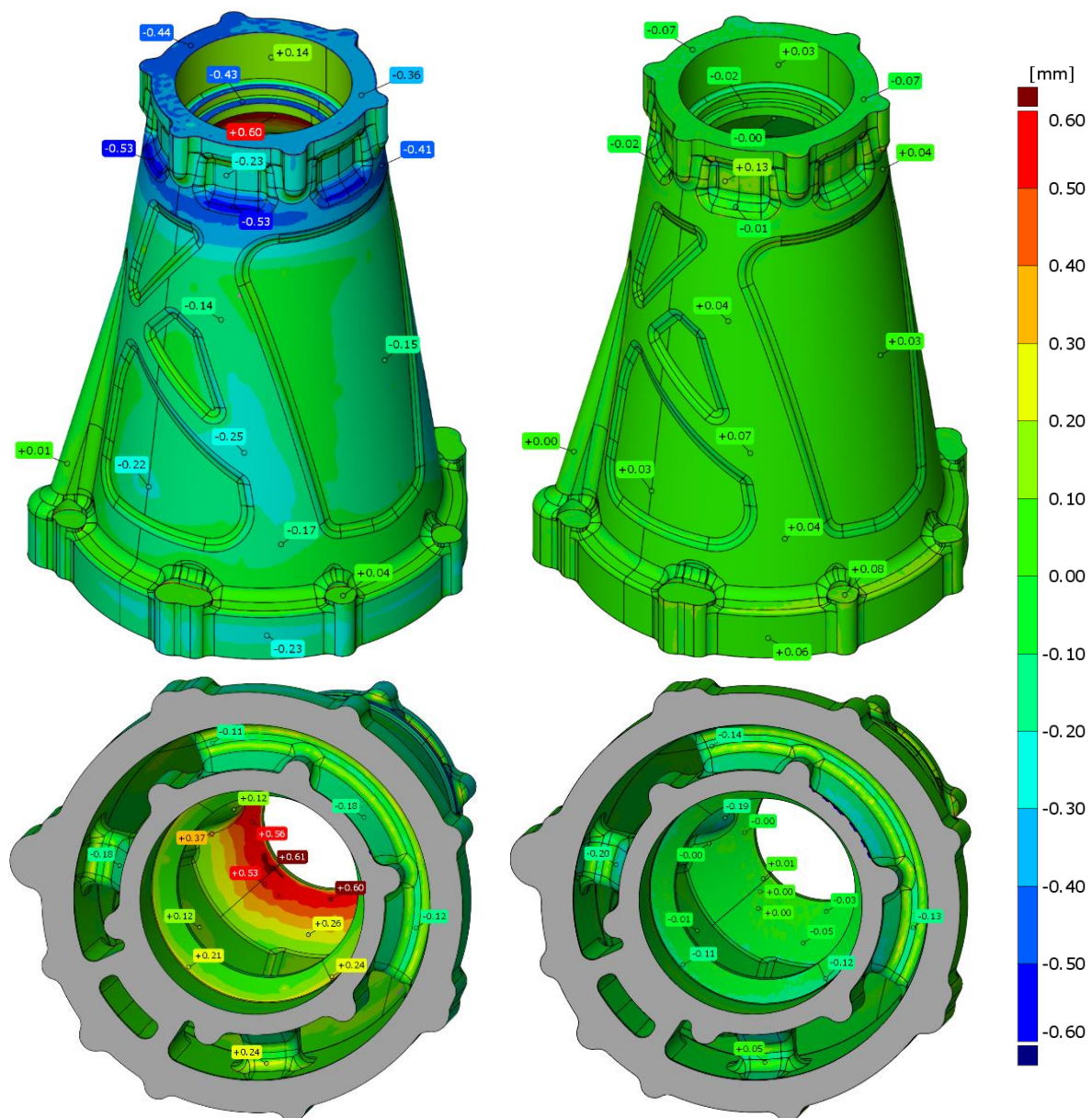


Рисунок 6.5 – Результат изготовления предеформированной модели

Подход, описанный в данной работе, может найти применение не только в технологии SLM, но и в других аддитивных и традиционных технологиях. На данный момент ведутся экспериментальные исследования по усовершенствованию созданной методики компенсации технологических деформаций, путем доработки архитектуры алгоритмов машинного обучения и ее тестирования на деталях из разных областей промышленности.

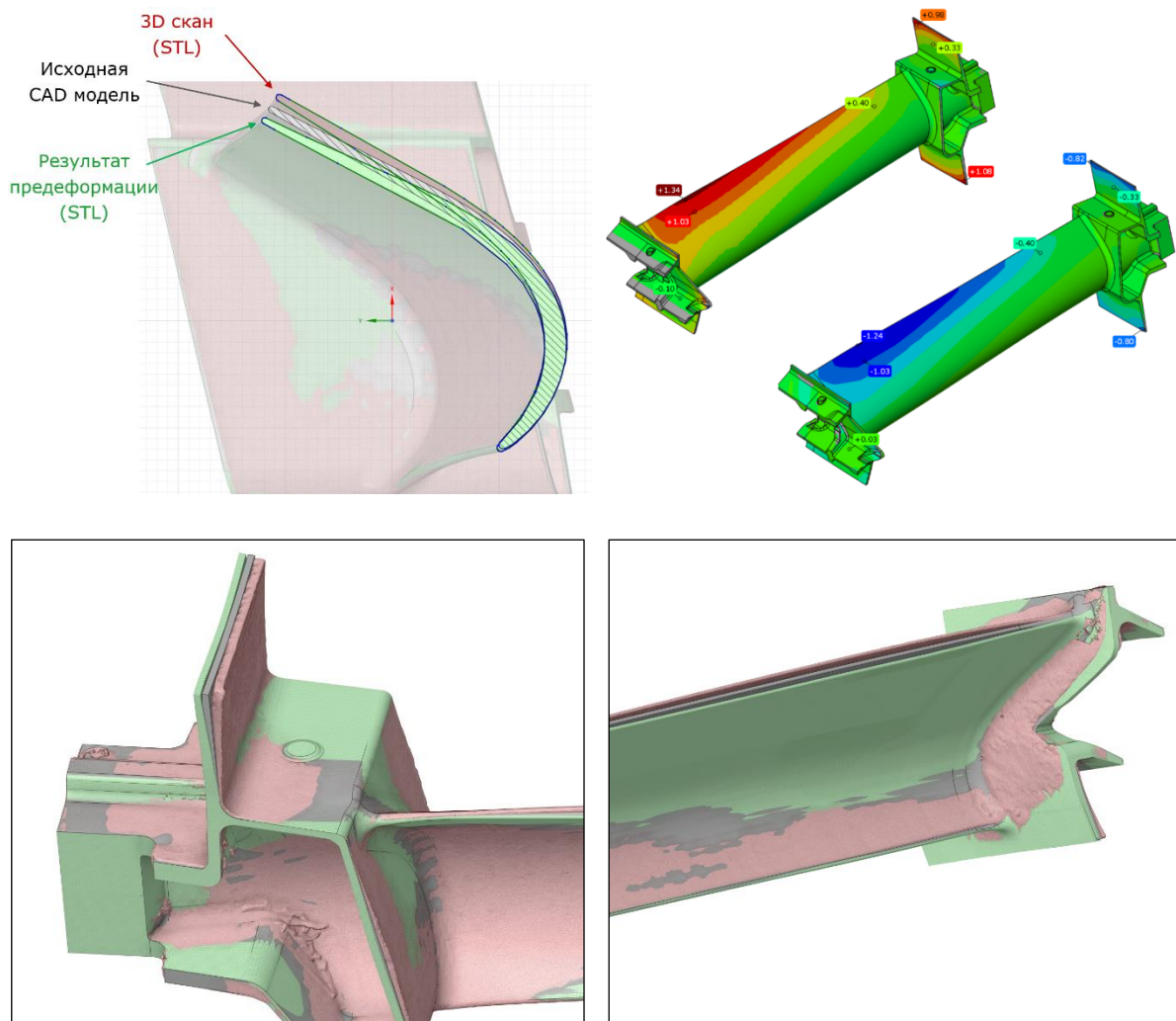


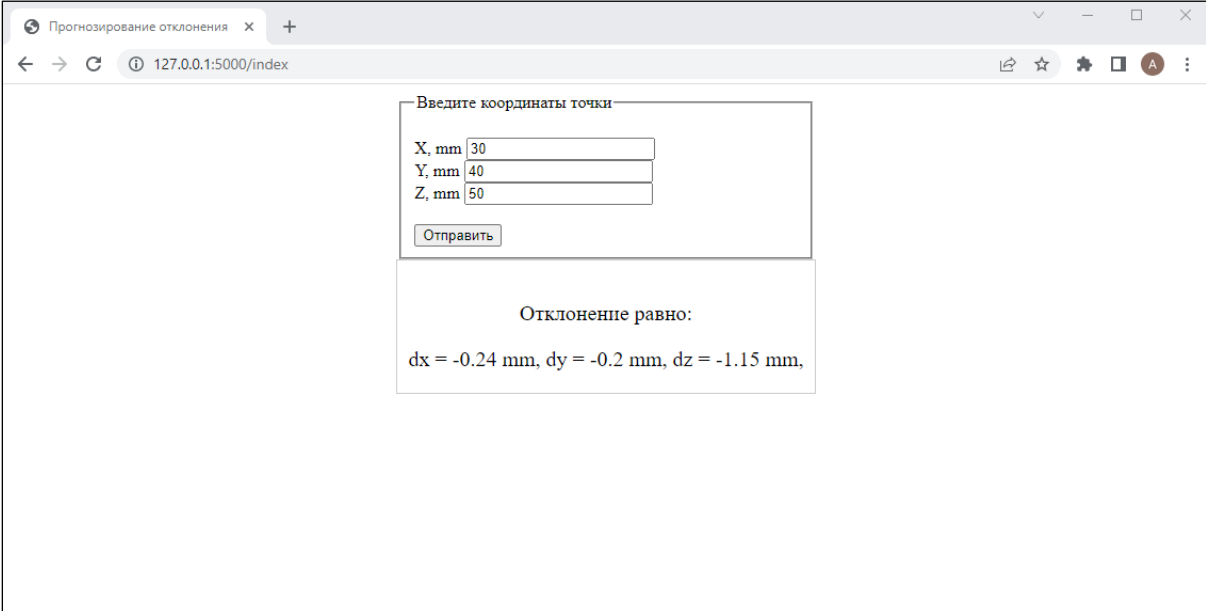
Рисунок 6.6 – Результат предеформации модели

Список используемой литературы

1. Андерсон, Карл Аналитическая культура. От сбора данных до бизнес-результатов / Карл Андерсон; пер. с англ. Юлии Константиновой; [науч. ред. Руслан Салахияев]. — М.: Манн, Иванов и Фербер, 2017. — 336с.
2. Аллен Б. Дауни – Основы Python. Научитесь думать как программист / Аллен Б. Дауни ; пер. с англ. С. Черникова ; [науч. ред. А. Родионов]. — Москва : Манн, Иванов и Фербер, 2021. — 304 с.
3. Билл Любанович. Простой Python. Современный стиль программирования. — СПб.: Питер, 2016. — 480 с.
4. Грас, Джоэл. Data Science. Наука о данных с нуля: Пер. с англ. - 2-е изд., перераб. и доп. - СПб.: БХВ-Петербург, 2021. - 416 с.
5. Григорьянц А.Г., Шиганов И.Н., Мисюров А.И., Третьяков Р.С. Лазерные аддитивные технологии в машиностроении. Москва, издательство МГТУ им. Н.Э. Баумана, 2018. 278 стр.
6. Джулли, Пал: Библиотека Keras - инструмент глубокого обучения / пер. с англ. А. А. Слинкин.- ДМК Пресс, 2017. – 249 с.
7. Жерон, Орельен. Прикладное машинное обучение с помощью Scikit-Learn и TensorFlow: концепции, инструменты и техники для создания интеллектуальных систем. Пер. с англ. - СПб.: ООО "Альфа-книга": 2018. - 688 с.
8. Митчелл, Райан. Современный скрапинг веб-сайтов с помощью Python. 2-е межд. издание . — СПб.: Питер, 2021.
9. Николенко С., Кадурын А., Архангельская Е. Глубокое обучение. Погружение в мир нейронных сетей. - СПб.: Питер. - 2020. - 480 с. ISBN: 978-5-4461-1537-2
10. Рассел С., Норвиг П. Искусственный интеллект: современный подход, 2-е изд.: Пер. с англ. - М. : Издательский дом “Вильямс”, 2007. - 1408 с.
11. Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы. - М.: Горячая Линия - Телеком. - 2013. - 384 с. ISBN: 978-5-9912-0320-3
12. Фостер Д.. Генеративное глубокое обучение. Творческий потенциал нейронных сетей. - СПб.: Питер. - 2020. - 336 с. - ISBN: 978-5-4461-1566-2
13. Шитиков В.К., Мастицкий С.Э. (2017) Классификация, регрессия и другие алгоритмы Data Mining с использованием R. 351 с.

Приложение 1. Разработка приложения

Внешний вид приложения выглядит следующим образом:



The screenshot shows a web browser window with the title "Прогнозирование отклонения". The address bar shows the URL "127.0.0.1:5000/index". The main content area contains a form with the heading "Введите координаты точки". Below the heading are three input fields labeled "X, mm", "Y, mm", and "Z, mm", with values 30, 40, and 50 respectively. A button labeled "Отправить" is located below the input fields. Below the form, the text "Отклонение равно:" is displayed, followed by the calculated values: "dx = -0.24 mm, dy = -0.2 mm, dz = -1.15 mm,".

Пользователь вводит координаты точки, в которой требуется определить отклонения. Приложение выводит предсказанные значения отклонений по трем направлениям, которые вычисляются по разработанной модели нейронной сети.