Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский университет ИТМО»
(Университет ИТМО)

Факультет цифровых трансформаций

Образовательная программа Финансовые технологии анализа данных

Направление подготовки (специальность) 11.04.02 Инфокоммуникационые технологии и системы связи

О Т Ч Е Т

о Научно-исследовательской практике

Тема задания: Создание алгоритма прогнозирования значения индикатора состояния пациента в следующем периоде по информации о текущем и предыдущих состояниях

Обучающийся Данилов Сергей Дмитриевич, J4112

Согласовано:
Руководитель практики от университета: Деревицкий Илья Владиславович, Научный сотрудник исследовательского центра в сфере искусственного интеллекта «Сильный искусственный интеллект в промышленности»

Практика пройдена с оценкой **5**

Дата **23.01.2023**

Санкт-Петербург
2023

Оглавление

Введение

Недавняя пандемия COVID-19 потрясла системы здравоохранения во всем мире. Большое количество пациентов и необходимость принимать решения по лечению в короткие сроки создали большую проблему для врачей по всему миру. Появилась необходимость в создании интеллектуальных систем [1], помогающих медицинским специалистам обработать большие массивы информации о существующем опыте борьбы с заболеванием, спланировать процесс лечения новых пациентов в крайне ограниченный срок, проверить важные гипотезы по развитию заболевания и так далее. Такие методы позволили бы быть готовым к возникновению эпидемий новых болезней, где специалисты обладали бы моделями прогнозирования развития заболевания и были бы способны принимать решения за меньший срок и с минимальным риском осложнения или летального исхода. К счастью, современные методы анализа данных обладают потенциалом в решении поставленных проблем. В последние годы машинное обучение и его подвид глубокое обучение стали популярным и надежным аналитическим методом в различных сферах деятельности, особенно в медицине. Большой объем накопленной медицинской информации за время пандемии и развитие архитектур моделей искусственного интеллекта образуют благоприятную почву к исследованиям в данной области. Так, в сфере клинических исследований разработаны алгоритмы анализа изображений компьютерной томографии, обработки последовательностей значений состояния пациента за время лечения. Аналогичные исследования не прекращаются и по прошествии активной фазы пандемии COVID-19, необходимо улучшать качество существующих моделей, изучать слабые места существующих траекторий лечения с помощью методов ИИ, решать новые задачи для других заболеваний, таких как вирус гриппа, онкологические заболевания. Такой интерес к созданию методов интеллектуального анализа медицинских данных делает исследования в данной области особо актуальными. В рамках научно – исследовательской работы мы остановимся на такой задаче, как предсказание

временных рядов параметров состояния пациента. Такие параметры включают в себя температуру тела, процентное содержание лимфоцитов в крови, артериальное давление и так далее. Разработка методов оценки возможных значений этих параметров позволила бы врачам и исследователям более детально понимать возможные пути развития заболевания и более осмысленного подходить к подбору методов лечения.

Итак, цель данного исследования – создание алгоритма прогнозирования значения индикатора состояния пациента в следующем периоде по информации о текущем и предыдущих состояниях, а также по выбранной стратегии лечения. Выполнение цели научно-исследовательской практики включает в себя процесс анализа существующих методов и проведение экспериментов применения моделей на реальных данных. Детальный список задач практики:

1. Провести поиск методов предсказания временных рядов
2. Проанализировать датасет COVID пациентов стационара НИМЦ Алмазова. Построить и обучить модели:
    a. рекуррентных нейронных сетей разных видов архитектур
    b. с помощью AutoML фреймворка Fedot.
3. Собрать данные об эффективности обученных моделей для каждого из параметров больного по метрикам MSE и R-квадрат
4. Подготовить отчет о проведенных экспериментах с таблицами, графиками и обсуждением полученных результатов

После выполнения всех пунктов плана научно-исследовательской практики ожидается скрипт на языке Python, который должен принимать информацию о текущем и предыдущих состояниях пациента, и с помощью обученной модели машинного обучения предсказывать значения параметров жизнедеятельности в следующем периоде.

Для проведения данного исследования был использован датасет с историей болезни пациентов, проходивших лечение от COVID-19 в НИМЦ Алмазова с мая

2020 года. Таблица содержит еженедельные записи о 1992 пациентах, где для каждой недели определены значения параметров жизнедеятельности, применяемые медикаменты, проведенные процедуры и общая информация о пациенте. В датасете (Таблица 1.) присутствует 53 колонки, которые включают в себя:

- Информацию о использовании 29 медикаментов
- 13 показателей состояния пациента
- 4 вида процедур
- Возраст пациента, длительность и результат лечения

		Температура_dinam_fact	Лимфоциты#_dinam_fact	АСТ_dinam_fact	ЧСС_dinam_fact
case	**t_point**				
GACAAcY	0	36.7	1.93	27.90	74.0
	1	36.3	2.00	25.70	71.0
	2	36.6	2.13	32.00	NaN
	3	36.5	2.12	27.00	71.0
GACAAnf	0	NaN	1.21	20.70	55.0
...	...	...	...	...	...
GACBB;O	0	36.9	1.22	22.34	54.0
	1	36.5	1.65	244.03	NaN
GACXAAB	0	36.5	1.11	21.80	NaN
	1	36.9	1.60	60.80	75.0
	2	36.6	3.45	17.20	NaN

Таблица 1. Часть датасета истории заболвеания COVID-19 пациентов НИМЦ Алмазова

Подробное описание параметров датасета представлено в Приложении.

Литературный обзор

На этапе обзора существующих методов я провел поиск научных статей по тематикам анализа временных рядов параметров состояния пациента и предсказания различных факторов на их основе.

Предикторами показателей жизнедеятельности авторы статьи (Ahmed Youssef Ali Amer et. al, 2021) [2] выбрали частоту сердечных сокращений, частоту дыхания и насыщение кислородом. Для прогнозирования этих параметров в следующих периодах было предложено использовать модель (Рисунок 1) локального обучения SVM, основанном на поиске методом k ближайших соседей (kNN-LS-SVM).

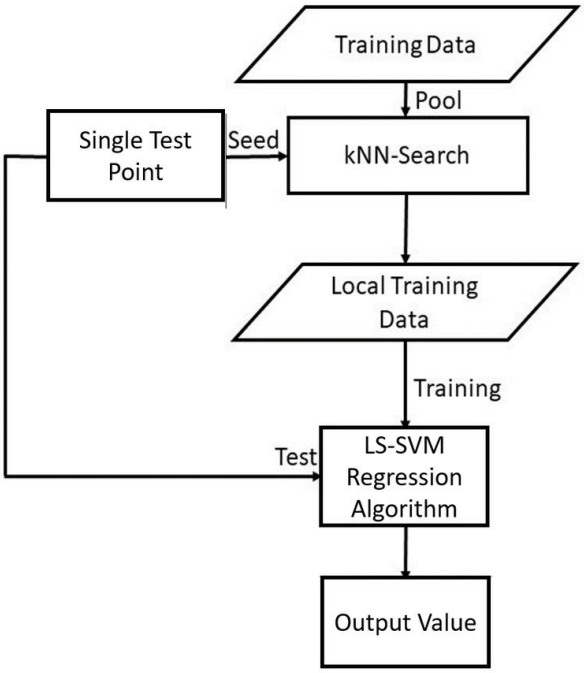

Рисунок 1. kNN-LS-SVM (Ahmed Youssef Ali Amer, 2021) [2]

Результаты работы модели таковы: при предсказании значений параметров в течение 3 часов модель совершила ошибку по метрике mAPE в 12%, 5% и 21.4% для вышеперечисленных параметров. Нужно обратить внимание на то, что метод, предложенный в данной статье, помогает оценить возможные границы значения параметра только для горизонта в 3 часа. Предсказание с таким

горизонтом может предотвратить внезапное ухудшение состояния, но для прогнозирования результата лечения, например, через неделю модель не сможет дать точный прогноз. Кроме того, параметры, используемые в обучении, снимались носимыми устройствами с большой частотой, поэтому эта модель может не показать такой точности для лабораторных тестов, которые обычно осуществляются не чаще недели.

Большинство исследователей акцентируют внимание на предсказание летального исхода процесса лечения или перевод пациента в отделение интенсивной терапии [3-11]. Одна из таких моделей (Farah E. et al., 2021) [3] предсказывает ухудшение состояния пациента в ближайшие 3–24 часа. Модель последовательно обрабатывает показатели жизнедеятельности пациента за предыдущие 24 часа: насыщение кислородом, частоту сердечных сокращений и температуру. Архитектура нейросети включает в себя слой рекуррентной сети LSTM, в которой обрабатываются последовательные данные о лечении, и полносвязный слой нейронов, принимающий на вход статичные данные о пациенте (Рисунок 2).

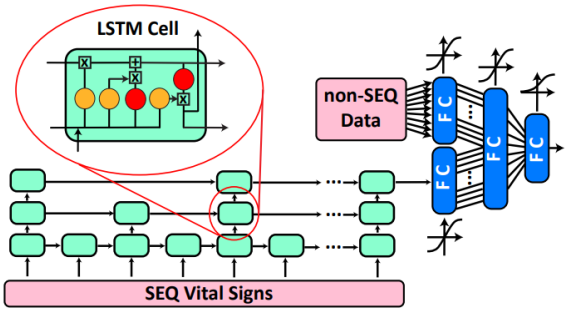

Рисунок 2. Модель LSTM + FC [3]

Особенно важно отметить процесс обучения данной нейросети, примененный авторами. На первом этапе LSTM нейросеть была отдельна обучена только на последовательных данных, а FC слой только на общей информации о пациенте. Далее веса, откалиброванные на первом этапе, применяются при обучении всей нейросети в целом. Такой метод позволяет сэкономить время при обучении сложных сетей и обеспечивает высокую сходимость модели.

В работе (Cristobal Esteban et al., 2016) [4] исследуется использование рекуррентных сетей в задаче прогнозирования состояния пациента. Авторами представлен подход, основанный на RNN, специально разработанный для клинической области, который сочетает в себе статическую и динамическую информацию для прогнозирования будущих событий. Цель модели состоит в том, чтобы предсказать исход пересадки почки, основываясь на информации, записанной в электронной медицинской карте каждого пациента. По произведенным экспериментам Gated Recurrent Units обеспечивает наилучшую производительность для этой задачи на горизонте прогнозирования от 6 до 12 месяцев.

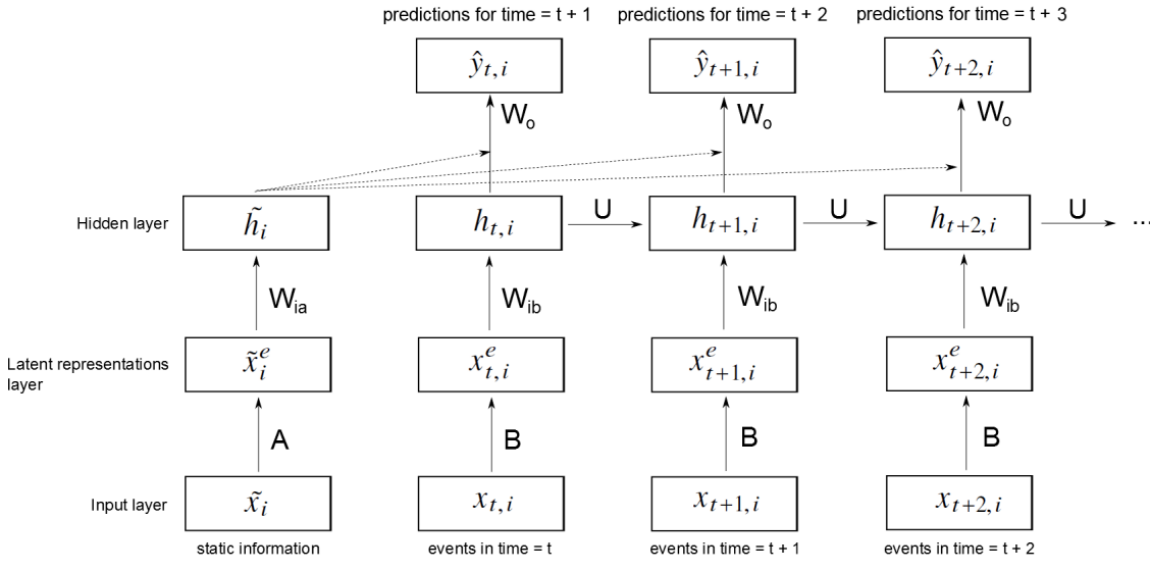

Рисунок 3. Схема модели с PCA методом [4]

При прогнозировании исхода или достижения какого-либо опасного состояния пациента важно предсказать его на ранних этапах. Руководствуясь этим принципом, авторы статьи (Junyi Gao, 2022) [5] сравнивают классические модели разных видов в задаче предсказания дней до окончания лечения и исхода лечения. Измерения производились на двух датасетах с помощью кросс

валидации. В качестве целевого значения использовался предложенный авторами метод OSMAE (Рисунок 4).

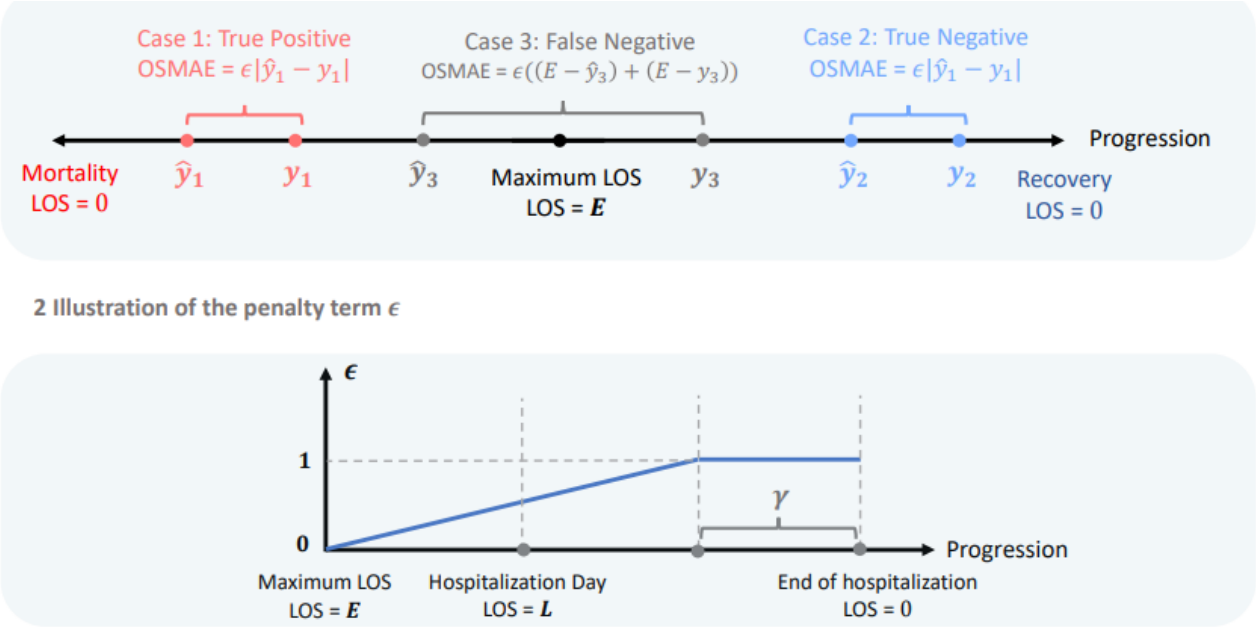

Рисунок 4 Метрика OSMAE [5]

Такой подход позволяет обучить модель не только правильно предсказывать исход заболевания, но и делать это как можно раньше, на минимальном количестве значений временного ряда. Лучшими методами предсказания из классических ML и DL моделей (по метрикам MSE, OSMAE) оказались Random Forest Regression, Gradient Boosting Decision Trees, RNN (LSTM, GRU) и TCN – Time Convolution Network.

Как мы видим из научных статей, самым популярным методом анализа последовательности являются рекуррентные нейронные сети. RNN добавляют память к искусственным нейронным сетям, но реализуемая память получается короткой — на каждом шаге обучения информация в памяти смешивается с новой и через несколько итераций полностью перезаписывается. Продвинутым методом запоминания информации во время обучения являются LSTM-модули [12], которые разработаны специально, чтобы избежать проблемы долговременной зависимости, запоминая значения как на короткие, так и на

длинные промежутки времени. Это объясняется тем, что LSTM-модуль не использует функцию активации внутри своих рекуррентных компонентов. Таким образом, хранимое значение не размывается во времени и градиент не исчезает при использовании метода обратного распространения ошибки во времени при тренировке сети. Ключевые компоненты LSTM-модуля: состояние ячейки и различные фильтры (Рисунок 5). О состоянии ячейки можно говорить, как о памяти сети, которая передает соответствующую информацию по всей цепочке модулей. Таким образом, даже информация из ранних временных шагов может быть получена на более поздних, нивелируя эффект кратковременной памяти. Немного усовершенствованный подход предлагается в модели GRU [14] (Gated Recurrent Units), где используются ворота обновления и сброса для решения проблемы исчезающего градиента стандартной RNN.

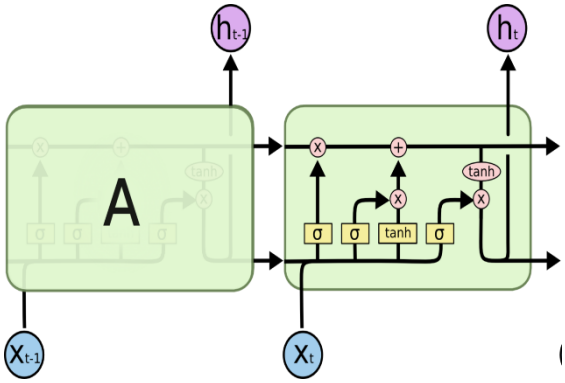

Рисунок 5. Структура LSTM [13]

AutoML Fedot. В настоящее время особую популярность набирают методы автоматической генерации пайплайнов моделей машинного обучения. Такой метод позволяет сократить время на предобработку данных, эксперименты с подбором нужной модели. Одним из таких фреймворков является библиотека Fedot [15], разработанная в университете ИТМО. Fedot кроме задач классификации и регрессии также осуществляет прогнозирование значений временного ряда [17]. Более того, фреймворк может искать зависимости сразу в нескольких последовательностях [18], например позволяет обрабатывать зависимости между рядами величин разных параметров состояния пациента.

Поэтому данный метод может подойти в задаче предсказания значений параметров состояния в истории пациента.

Основной акцент в работе фреймворка делается на относительно сложное управление взаимодействиями между различными вычислительными блоками пайплайнов. В первую очередь, это касается этапа непосредственного создания модели машинного обучения. FEDOT позволяет не просто подобрать лучший вариант модели и обучить ее, а создать сложную (композитную) модель.. Для заданного набора данных строится решение в виде графа (DAG), узлы которого представлены моделями МО, процедурами предобработки и трансформации данных. Структура этого графа, а также параметры каждого узла и подвергаются обучению [16]. Готовая модель сохраняется в виде json структуры и может повторно использоваться на тестовых данных.

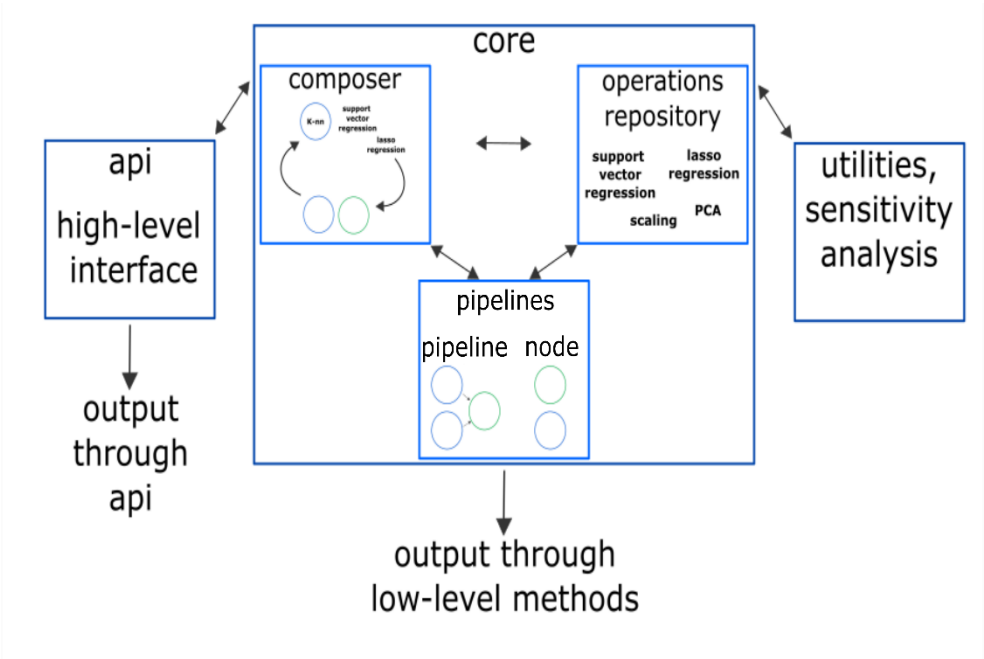

Рисунок 6. Принцип работы фреймворка Fedot [16]

Описание исследований

1. *Предобработка данных*

Перед обучением моделей машинного обучения я провел предобработку табличных данных. Так как лабораторные анализы пациента имеют разный масштаб, для корректного обучения модели следует нормировать значения. Для этого я использовал метод MinMaxScaler из библиотеки sklearn. При обучении моделей не используется информации о исходе и продолжительности лечения, поэтому данные колонки были отброшены совсем.

Проблемой структуры данных являлись пропущенные значения некоторых параметров. Было несколько вариантов решения

- Отбросить строчки с пропущенными значениями
- Заполнить предыдущим или средним значением между предыдущим и следующим измерением
- Применить регрессию на пропущенное значение методом MICE

Я решил использовать третью стратегию, которая реализована в методе Iterative Imputer библиотеки sklearn [19]. Алгоритм делает это через повторяющийся циклический метод: на каждом шаге столбец, где отсутствует значение, обозначается как выход модели y, а другие столбцы признаков обрабатываются как входы X, по которым простая модель регрессии (я использовал DecisionTreeRegressor) находит значение пропущенного элемента. Порядок действий проделывается для каждого признака итеративно до указанного количества повторений или предопределенного критерия останова цикла.

Модели в данной работе должны принимать определенное количество векторов значений – состояний пациентов в текущий и предыдущие периоды. Для этого я создал метод, формирующий матрицу из историй заболевания с количеством записей определенной длины. На рисунке 7 показан пример такого разделения для окна размером три недели.

В качестве предикторов выступают все 47 параметров стратегии лечения и состояния пациента в текущей неделе и предыдущих в истории заболевания, а

таргетами являются векторы из 13 параметров состояния пациента в следующей неделе.

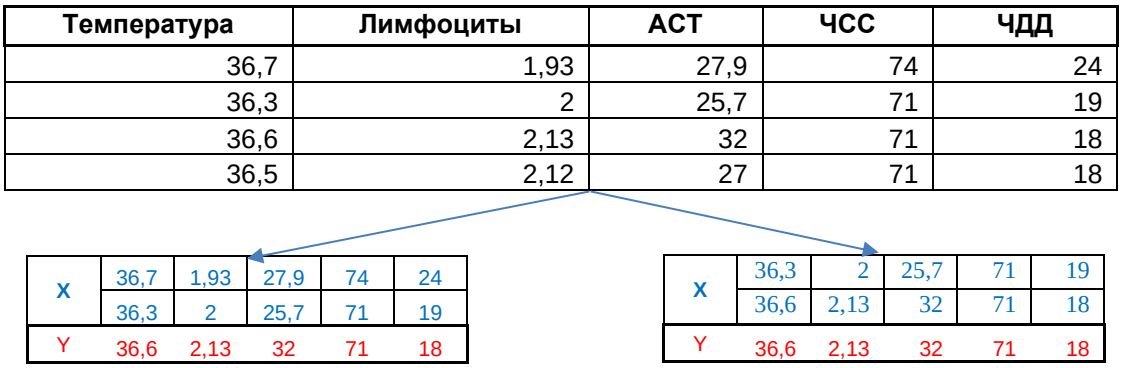

Рисунок 7 Разделение временного ряда на окна

В рамках данного исследования я использовал окна входных значений длины от 1 вектора (только текущие значения параметров) до 4 (текущие значения и 3 вектора предыдущих недель). Подготовленные матрицы X и Y я разделил в соотношении 80:10:10 для обучения, теста и валидации модели соответственно. Для оценки эффективности обученных моделей в ходе обучения я использовал метрики MSE (среднеквадратичная ошибка) и R^2 (коэффициент детерминации). Эксперименты я провел отдельно для окон разных длины чтобы определить, какие модели лучше использовать для предсказания параметра при различной продолжительности лечения.

2. *Обучение моделей*

Для построения моделей я использовал библиотеку Keras [20]. Сперва я использовал модель простого многослойного перцептрона (MLP) для предсказания параметров, когда у нас имеется только одна запись в истории заболевания. В качестве функции активации для скрытых слоев я использовал relu, а на выходе модели использовалась линейная функция активации. Для описания и тренировки модели я использовал слои Flatten и Dense библиотеки keras. Были проведены несколько экспериментов с различным количеством

скрытых слоев нейронной сети, количеством недель в качестве входных параметров, размерами пакетов.

Общую архитектуру модели можно увидеть на рисунке 8

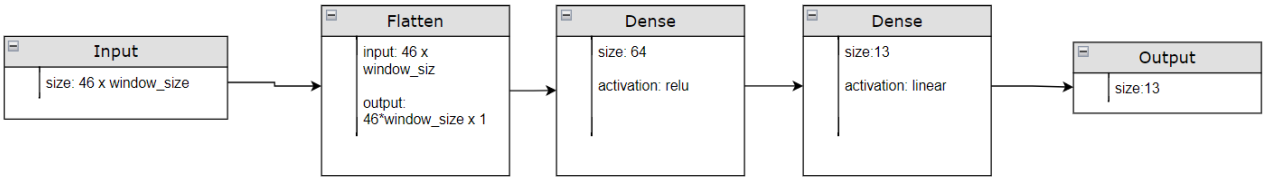

Рисунок 8. Multi-Layer Perceptron

Обучение модели прошло успешно, переобучения не выявлено. Результаты можно увидеть в таблице:

Параметр	Модель	R2	MSE
Температура_dinam_fact	Flatten5Layers_batch4	0.010550	0.002376
Лимфоциты#_dinam_fact	Faltten1Layer	0.824537	0.001349
АСТ_dinam_fact	Faltten1Layer	0.449263	0.000157
ЧСС_dinam_fact	Flatten3Layers_batch8	0.010583	0.008367
ЧДД_dinam_fact	Flatten5Layers_batch4	0.344241	0.000147
Билирубин общий_dinam_fact	Flatten3Layers	0.397709	0.001319
MPV- Средн.объем тромбоцитов_dinam_fact	Flatten5Layers_batch4	0.652261	0.005411
PCT- Тромбокрит_dinam_fact	Faltten1Layer	0.314895	0.008465
Лимфоциты%_dinam_fact	Faltten1Layer	0.581243	0.009657
снижение_сознания_dinam_fact	Flatten5Layers_Mixed_batch4	0.910550	0.005053
Степень тяжести по КТ_dinam_fact	Flatten5Layers	0.037550	0.067288
Лактатдегидрогеназа_dinam_fact	Faltten1Layer	0.191407	0.000576
PDW- Индекс расп.по объему тр._dinam_fact	Flatten4Layers	0.642589	0.004587

Таблица 2. Лучшие результаты моделей MLP для окна длиной 1

3. RNN нейросети

Далее я провел эксперименты с RNN нейросетями. Входные данные для модели включали в себя матрицу из векторов состояний, где в каждом векторе содержится все параметры для обучения – как динамические, так и статические. Каждый вектор последовательно передается нейронной сети, а вектор

последнего состояния сети RNN передается полносвязному слою. В результате получаем вектор из 13 значений – предсказаний параметров в следующей недели.

Один блок RNN обрабатывает последовательность и корректирует веса один раз. Я добавил еще один слой RNN после основного, который принимает векторы состояний предыдущей RNN. Так мы можем достичь большей сложности модели и позволить найти скрытые взаимосвязи в данных.

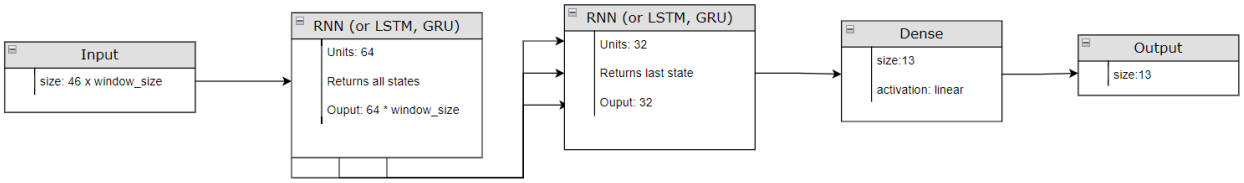

Рисунок 9. Схема модели с двухслойным RNN

Также я провел эксперименты с двумя типами входных данных – когда передается кортеж из динамических данных и вектор данных о медикаментах, процедурах и о пациенте.

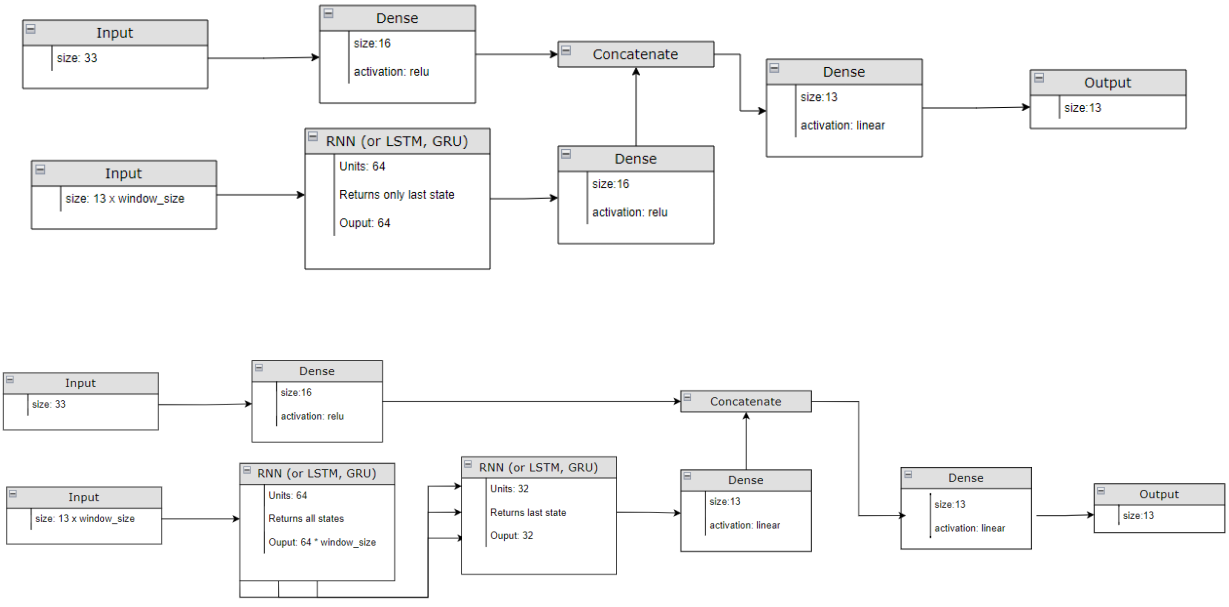

Рисунок 10. Схема моделей с дополнительным входом.

Для такого подхода проведены аналогичные эксперименты с количеством нейронов состояний, размерами пакетов.

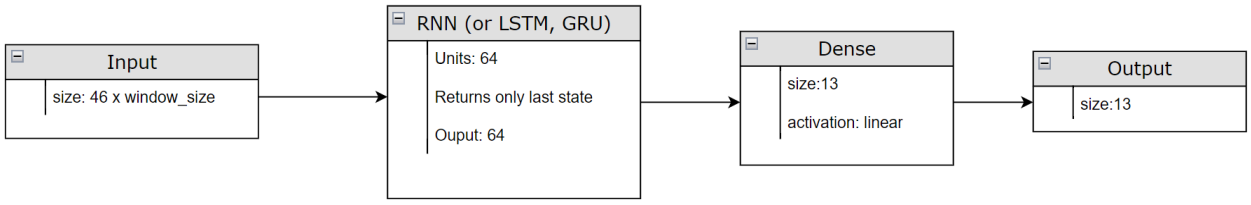

Рисунок 11. Схема модели с RNN

В качестве реализаций RNN я использовал SimpleRNN, LSTM и GRU. Были проведены эксперименты с различными длинами вектора состояния RNN – 13, 32, 64. Кроме этого, были использованы разные объемы пакетов входных данных – 1, 2, 4, 6, 8, 12. В колонке «Модель» описана конфигурация модели –

- LSTM13_Batch8 – сеть с 13 нейронами состояния, размер пакета - 8
- LSTMxLSTM – два слоя LSTM
- StaticDense_DynamicLSTM – модель с двумя входами

Результаты экспериментов представлены в таблице 2:

Параметр	Модель	R2	MSE
Температура_dinam_fact	GRU13_Batch8	0.024583	0.001804
Лимфоциты#_dinam_fact	StaticDense_DynamicLSTM_twoLayers	0.661245	0.001363
АСТ_dinam_fact	GRU64_GRU32_Dense	0.604499	0.000167
ЧСС_dinam_fact	RNNMixed	0.050058	0.007378
ЧДД_dinam_fact	LSTMtoLSTM_Mixed	0.413600	0.000145
Билирубин общий_dinam_fact	LSTM64Mixed	0.361337	0.000881
MPV- Средн.объем тромбоцитов_dinam_fact	GRU32xGRU32_Batch4	0.651195	0.004539
PCT- Тромбокрит_dinam_fact	GRU32xGRU32_Batch4	0.449788	0.004105
Лимфоциты%_dinam_fact	StaticDense_DynamicLSTM_twoLayers	0.596125	0.007296
снижение_сознания_dinam_fact	LSTM13_Batch4	0.997207	0.000108
Степень тяжести по КТ_dinam_fact	LSTM13_Batch4	0.015368	0.064500
Лактатдегидрогеназа_dinam_fact	GRU64_GRU13_Dense	0.431431	0.000361
PDW- Индекс расп.по объему тр._dinam_fact	LSTM13_Batch4	0.718983	0.003890

Таблица 3 Результаты моделей для окна в одну неделю и текущее состояние

Параметр	Модель	R2	MSE
Температура_dinam_fact	SimpleRNN13_Tanh	-0.025246	0.001267
Лимфоциты#_dinam_fact	LSTM64xLSTM32_Batch4	0.991304	0.000082
АСТ_dinam_fact	LSTM13_Batch2	0.701792	0.000217
ЧСС_dinam_fact	LSTM64xLSTM32_Batch4	-0.036390	0.011386
ЧДД_dinam_fact	LSTM13_Batch1	0.634257	0.000103
Билирубин общий_dinam_fact	LSTM64_LSTM32_Dense	0.283741	0.001486
MPV- Средн.объем тромбоцитов_dinam_fact	LSTM13_Batch8	0.721695	0.003346
PCT- Тромбокрит_dinam_fact	LSTM64xLSTM32_Batch4	0.195863	0.005125
Лимфоциты%_dinam_fact	LSTM64xLSTM32_Batch4	0.639118	0.006520
снижение_сознания_dinam_fact	LSTM64xLSTM13	0.995149	0.000445
Степень тяжести по КТ_dinam_fact	LSTM32	-0.044031	0.031057
Лактатдегидрогеназа_dinam_fact	LSTM64xLSTM32_Batch4	0.499811	0.000392
PDW- Индекс расп.по объему тр._dinam_fact	LSTM13_Batch4	0.656922	0.003762

Таблица 4. Результаты для окна размером 2 недели

Параметр	Модель	R2	MSE
Температура_dinam_fact	GRU13_Batch1	0.017721	0.001471
Лимфоциты#_dinam_fact	GRU32_Batch4	0.602346	0.009426
АСТ_dinam_fact	LSTM64xLSTM13_Batch4	0.134370	0.000607
ЧСС_dinam_fact	LSTM13_Batch4	0.071622	0.012290
ЧДД_dinam_fact	LSTM13_Batch8	0.404697	0.000176
Билирубин общий_dinam_fact	GRU13_Batch4	0.322373	0.002310
MPV- Средн.объем тромбоцитов_dinam_fact	GRU32_Batch4	0.670607	0.003386
PCT- Тромбокрит_dinam_fact	GRU13_Batch2	0.411801	0.002133
Лимфоциты%_dinam_fact	GRU32xGRU32_Batch4	0.754267	0.005626
снижение_сознания_dinam_fact	StaticDense_DynamicLSTM_oneLayer	0.974094	0.002201
Степень тяжести по КТ_dinam_fact	LSTM32	0.200391	0.031859
Лактатдегидрогеназа_dinam_fact	GRU13_Batch2	0.290591	0.000667
PDW- Индекс расп.по объему тр._dinam_fact	StaticDense_DynamicGRU_TwoLayersResult	0.727560	0.003408

Таблица 5. Результаты для окна размером 3 недели

Параметр	Модель	R2	MSE
Температура_dinam_fact	GRU13_Batch8	0.024583	0.001804
Лимфоциты#_dinam_fact	StaticDense_DynamicLSTM_twoLayers	0.661245	0.001363
АСТ_dinam_fact	GRU64_GRU32_Dense	0.604499	0.000167
ЧСС_dinam_fact	RNNMixed	0.050058	0.007378
ЧДД_dinam_fact	LSTMtoLSTM_Mixed	0.413600	0.000145
Билирубин общий_dinam_fact	LSTM64Mixed	0.361337	0.000881
MPV- Средн.объем тромбоцитов_dinam_fact	GRU32xGRU32_Batch4	0.651195	0.004539
PCT- Тромбокрит_dinam_fact	GRU32xGRU32_Batch4	0.449788	0.004105
Лимфоциты%_dinam_fact	StaticDense_DynamicLSTM_twoLayers	0.596125	0.007296
снижение_сознания_dinam_fact	LSTM13_Batch4	0.997207	0.000108
Степень тяжести по КТ_dinam_fact	LSTM13_Batch4	0.015368	0.064500
Лактатдегидрогеназа_dinam_fact	GRU64_GRU13_Dense	0.431431	0.000361
PDW- Индекс расп.по объему тр._dinam_fact	LSTM13_Batch4	0.718983	0.003890

Таблица 6. Результаты для окна размером 4 недели

	Окно	Модель	R2	MSE
снижение_сознания_dinam_fact	2	LSTM13_Batch4	0.997207	0.000108
Лимфоциты#_dinam_fact	3	LSTM64xLSTM32_Batch4	0.991304	0.000082
Лимфоциты%_dinam_fact	4	GRU32xGRU32_Batch4	0.754267	0.005626
PDW- Индекс расп.по объему тр._dinam_fact	4	StaticDense_DynamicGRU_TwoLayersResult	0.727560	0.003408
MPV- Средн.объем тромбоцитов_dinam_fact	3	LSTM13_Batch8	0.721695	0.003346
АСТ_dinam_fact	3	LSTM13_Batch2	0.701792	0.000217
ЧДД_dinam_fact	3	LSTM13_Batch1	0.634257	0.000103
Лактатдегидрогеназа_dinam_fact	3	LSTM64xLSTM32_Batch4	0.499811	0.000392
PCT- Тромбокрит_dinam_fact	2	GRU32xGRU32_Batch4	0.449788	0.004105
Билирубин общий_dinam_fact	1	Flatten3Layers	0.397709	0.001319
Степень тяжести по КТ_dinam_fact	4	LSTM32	0.200391	0.031859
ЧСС_dinam_fact	4	LSTM13_Batch4	0.071622	0.012290
Температура_dinam_fact	2	GRU13_Batch8	0.024583	0.001804

Таблица 7. Лучшие результаты для нейросетей

4. Fedot фреймворк

Далее я воспользовался AutoML фреймворком Fedot для автоматического подбора и обучения моделей. Для построения пайплайнов моделей я использовал режим “regression”, где в качестве входных параметров я объединял данные нескольких недель и параметры пациента в один вектор. Предобработки на таких

данных не выполнялось, так как фреймворк сам подбирает лучшие методы для работы с такими данными. Получившиеся пайплайны для параметров можно увидеть на рисунках 11 и 12.

Пайплайн 1: Температура, ЧДД, ЧСС, Билирубин, Лактатдегидраза, PCT, MPV

Пайплайн 2: Снижение сознания, Лимфоциты, АСТ, тяжесть по КТ

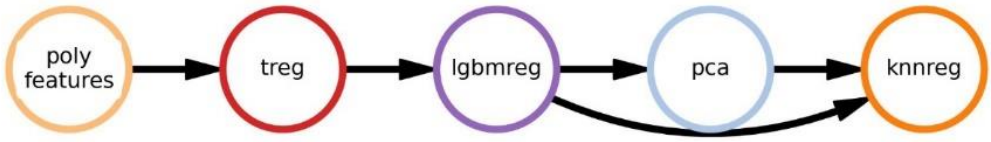

Рисунок 11. Пайплайн 1 – Дерево решений, градиентный бустинг, PCA, kNN

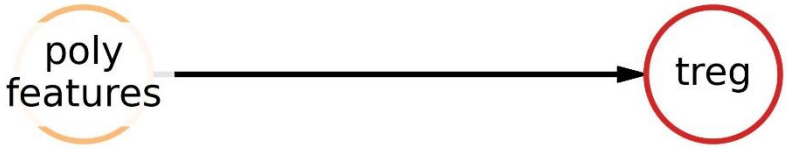

Рисунок 12. Пайплайн 2 – Регрессия с помощью дерева решений

Результаты регрессии для посторенных моделей для разных окон представлены в таблицах 8 и 9

window	1		2		3		4	
metric	MSE	R2	MSE	R2	MSE	R2	MSE	R2
Температура_dinam_fact	0.007989	0.273022	0.018461	0.154734	0.002140	0.751755	0.011051	0.382191
Лимфоциты#_dinam_fact	0.000109	0.969521	0.000438	0.942135	0.000037	0.995090	0.000540	0.970245
АСТ_dinam_fact	0.002917	0.083747	0.002458	0.572168	0.008054	0.137684	0.022607	-0.018419
ЧСС_dinam_fact	0.014933	0.046259	0.006575	0.619092	0.024282	0.248748	0.013490	0.473071
ЧДД_dinam_fact	0.073847	-25.489654	0.001959	0.529697	0.502955	-60.869972	0.014093	0.727775
Билирубин общий_dinam_fact	0.003615	0.542981	0.002330	0.648010	0.009495	0.796936	0.023661	0.572934
MPV- Средн.объем тромбоцитов_dinam_fact	0.004272	0.808650	0.003061	0.862927	0.005353	0.784558	0.005282	0.902197
PCT- Тромбокрит_dinam_fact	0.009543	0.495945	0.015439	0.478747	0.015793	0.511469	0.012522	0.770217
Лимфоциты%_dinam_fact	0.007285	0.683692	0.006144	0.756382	0.009009	0.529539	0.017245	0.547227
снижение_сознания_dinam_fact	0.004670	0.932692	0.007800	0.870089	331.193322	-4569.352326	0.000514	0.995699
Степень тяжести по КТ_dinam_fact	0.055066	0.181519	0.050018	0.150659	0.058817	0.323039	0.070524	0.264397
Лактатдегидрогеназа_dinam_fact	0.002481	0.179927	0.006425	0.569190	0.006259	0.852838	0.013885	0.832242
PDW- Индекс расп.по объему тр._dinam_fact	0.008379	0.858741	0.007526	0.848841	0.007140	0.896949	0.013307	0.784932

Таблица 8. Результаты фреймворка Fedot

Параметр	Окно	R2	MSE
Лимфоциты#_dinam_fact	1	0.973649	0.000067
PDW- Индекс расп.по объему тр._dinam_fact	3	0.918739	0.004195
MPV- Средн.объем тромбоцитов_dinam_fact	4	0.912909	0.004195
снижение_сознания_dinam_fact	2	0.906946	0.005766
Лактатдегидрогеназа_dinam_fact	4	0.906615	0.007804
ЧСС_dinam_fact	4	0.881666	0.003261
Билирубин общий_dinam_fact	4	0.867605	0.004866
PCT- Тромбокрит_dinam_fact	4	0.735104	0.014774
Лимфоциты%_dinam_fact	2	0.696388	0.006229
ЧДД_dinam_fact	4	0.681255	0.011515
Температура_dinam_fact	4	0.608237	0.015320
Степень тяжести по КТ_dinam_fact	3	0.342180	0.028965
АСТ_dinam_fact	1	0.181626	0.001513

Таблица 9. Лучшие результаты по параметрам при использовании Fedot

5. *Результаты*

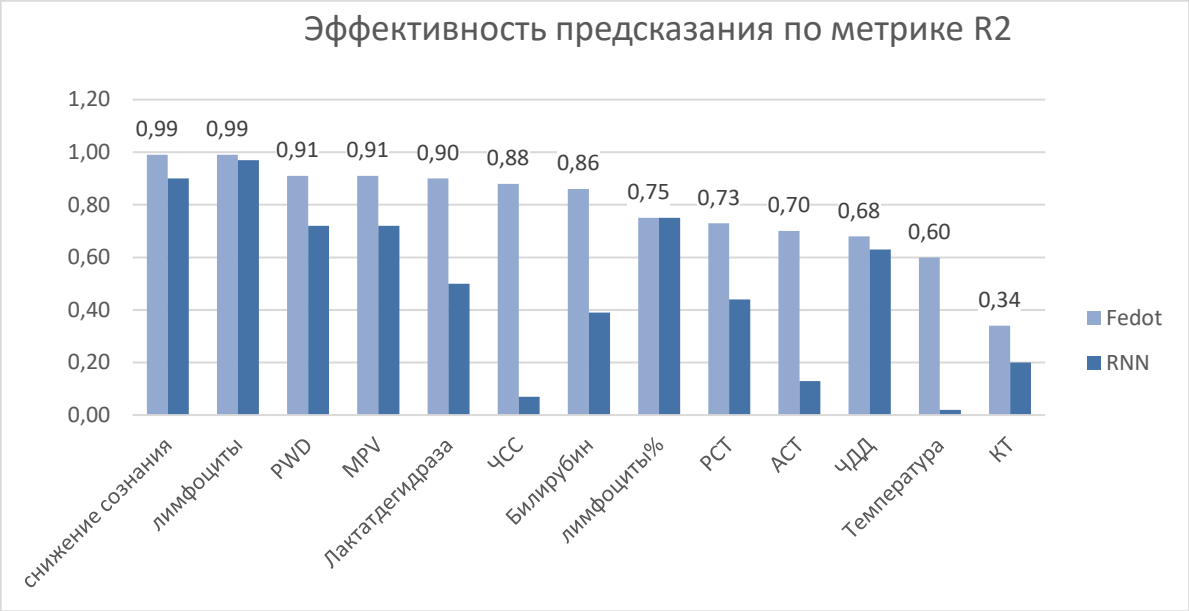

Рисунок 13. Эффективность предсказания параметров
с помощью нейросетей и пайплайнов Fedot по метрике объясненной дисперсии

- 11 параметров из 13 предсказываются моделями с метрикой $R^2 > 0.7$
- 8 из 13 параметров предсказываются с долей объяснённой дисперсии больше 0.5
- Рекуррентные сети LSTM и GRU подходят для предсказания значений параметров пациента. Добавление второго слоя рекуррентной нейросети увеличивает эффективность модели.
- Результат также улучшается при обучении на пакетах данных - «батчах». Лучшее количество сэмплов в батче по итогам экспериментов – 4.
- Добавление статичных данных не дает сильного улучшения метрик для многих параметров
- Более точные предсказания получаются при предсказании с окном в 3 записи о состоянии пациента
- Модели, построенные с помощью AutoML фреймворка Fedot показали лучший результат в сравнении с нейросетями, благодаря автоматической предобработки и более глубокому подбору моделей и параметров в фреймворке.

Заключение

В ходе практики было полностью выполнено Индивидуальное задание. Был проведен обзор методов анализа истории заболевания пациента, моделей машинного обучения для предсказания значений временных рядов. По результатам анализа был подготовлен список моделей, которые в дальнейшем были обучены на датасете с историями пациентов.

Планируемые результаты практики достигнуты полностью. Обученные в ходе практики модели регрессии дают хорошее качество предсказания по метрикам коэффициента детерминации и среднеквадратичной ошибки для большинства параметров. В результате экспериментов пришли выводу, что пайплайны моделей МО, построенные с помощью фреймворка Fedot показали наилучший результат. Рекуррентные нейронные сети LSTM и GRU с использованием двух слоев так же показали хороший результат. Эффективность моделей доказывает тот факт, что модели машинного обучения могут использоваться при прогнозировании изменения состояния факторов жизнедеятельности пациента. Код и дополнительные материалы исследований доступны на репозитории https://github.com/geniusserg/CovidPredict

Список использованных источников

[1] Onur Dogan et al., 2021, A systematic review on AI/ML approaches against COVID-19 outbreak, Complex Intelligent Systems Journal

[2] Ahmed Youssef Ali Amer et al., (2021) «Vital Signs Prediction for COVID-19 Patients in ICU», Sensors.

[3] Farah E. Shamout et al., (2022) «Deterioration Prediction using Time-Series of Three Vital Signs and Current Clinical Features Amongst COVID-19 Patients», Sensors

[4] Cristóbal Esteban et al., (2016) «Predicting Clinical Events by Combining Static and Dynamic Information Using Recurrent Neural Networks» BMC Medical Information Decision Making

[5] Junyi Gao et al., 2022. A Comprehensive Benchmark for COVID-19 Predictive Modeling Using Electronic Health Records in Intensive Care: Choosing the Best Model for COVID-19 Prognosis

[6] Sicen Lu et al. (2020) «A hybrid method of recurrent neural network and graph neural network for next-period prescription prediction» International Journal of Machine Learning and Cybernetics.

[7] Davi Silva Rodrigues et al. (2022) «Predicting the outcome for COVID-19 patients by applying time series classification to electronic health records» BMC Medical Informatics and Decision Making.

[8] Siavash Bolourani et al. (2021) «A Machine Learning Prediction Model of Respiratory Failure Within 48 Hours of Patient Admission for COVID-19: Model Development and Validation» Journal of Med Internet Res

[9] Luchen Liu et al. (2019) «Learning Hierarchical Representations of Electronic Health Records for Clinical Outcome Prediction» AMIA Annual Symposium Proceedings

[10] Farah E. Shamout et al., (2022) «Deterioration Prediction using Time-Series of Three Vital Signs and Current Clinical Features Amongst COVID-19 Patients» Sensors

[11] Chenxi Sun et al., (2021) « Predicting COVID-19 disease progression and patient outcomes based on temporal deep learning » BMC Medical Information Decision Making

[12] Sepp Hochreiter, 1997. Long Short-term Memory

[13] Christopher Olah, 2015, Understanding LSTM Networks. URL: http://colah.github.io/posts/2015-08-Understanding-LSTMs

[14] Cho et al., 2014. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches.

[15] Nikolay O. Nikitin et al, 2021, Automated Evolutionary Approach for the Design of Composite Machine Learning Pipelines

[16] Mikhail Sarafanov, 2021. How AutoML helps to create composite AI? URL: https://towardsdatascience.com/how-automl-helps-to-create-composite-ai-f09e05287563

[17] Mikhail Sarafanov, 2021. AutoML for time series: advanced approaches with FEDOT framework. URL: https://towardsdatascience.com/automl-for-time-series-advanced-approaches-with-fedot-framework-4f9d8ea3382c

[18] Mikhail Sarafanov, 2022. What to Do If a Time Series Is Growing (But Not in Length). URL: https://towardsdatascience.com/what-to-do-if-a-time-series-is-growing-but-not-in-length-421fc84c6893

[19] Iterative Imputation. Scikit-learn official documentation. URL: https://scikit-learn.org/stable/modules/generated/sklearn.impute.IterativeImputer.html

[20] Keras official API documentation. URL: https://keras.io/api

Приложение

Список параметров в истории COIVD-19 пациентов НИМЦ Алмазова

1. Медикаменты:
    1. *омепразол_stat_control*
    2. *надропарин кальция_stat_control*
    3. *эзомепразол_stat_control*
    4. *амлодипин_stat_control*
    5. *амброксол_stat_control*
    6. *домперидон_stat_control*
    7. *меброфенин_stat_control*
    8. *технеций_stat_control*
    9. *мометазон_stat_control*
    10. *бисопролол_stat_control*
    11. *дексаметазон_stat_control*
    12. *гидрохлоротиазид_stat_control*
    13. *гидроксихлорохин_stat_control*
    14. *рабепразол_stat_control*
    15. *эноксапарин натрия_stat_control*
    16. *периндоприл_stat_control*
    17. *ацетилцистеин_stat_control*
    18. *азитромицин_stat_control*
    19. *валсартан_stat_control*
    20. *метилпреднизолон_stat_control*
    21. *лоратадин_stat_control*
    22. *хлорохин_stat_control*
    23. *натрия хлорид_stat_control*
    24. *индапамид_stat_control*
    25. *преднизолон_stat_control*
    26. *аторвастатин_stat_control*
    27. *декстран_stat_control*
    28. *лизиноприл_stat_control*
    29. *лозартан_stat_control*
2. Показатели состояния
    1. *Температура_dinam_fact*
    2. *Лимфоциты#_dinam_fact*
    3. *АСТ_dinam_fact2*
    4. *ЧСС_dinam_fact*
    5. *ЧДД_dinam_fact*
    6. *Билирубин общий_dinam_fact*
    7. *MPV- Средн.объем тромбоцитов_dinam_fact*
    8. *PCT- Тромбокрит_dinam_fact*
    9. *Лимфоциты%_dinam_fact*
    10. *снижение_сознания_dinam_fact*

11. *Степень тяжести по КТ_dinam_fact*
12. *Лактатдегидрогеназа_dinam_fact*
13. *PDW- Индекс расп.по объему тр._dinam_fact*

3. Возраст пациента
    1. *age_stat_fact*
4. Процедуры
    1. *трансфузия_dinam_control*
    2. *оксигенотерапия_dinam_control*
    3. *НИВЛ_dinam_control*
    4. *ИВЛ_dinam_control*
5. Длительность и результат лечения
    1. *long_observation_tar* – длительность в днях
    2. *outcome_tar* – исход лечения
    3. *process_stages* – фаза эпидемии
    4. *current_process_duration* – количество дней с начала заболевания
    5. *admission_date* – дата поступления пациента в стационар
    6. *end_epizode* – является ли запись в истории последней