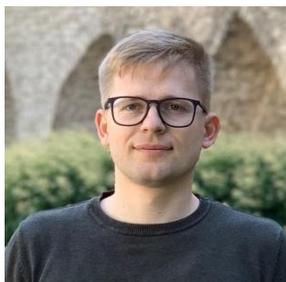


УДК [611.018.51+615.47]:612.086.2

## СПОСОБЫ МАШИННОГО ОБУЧЕНИЯ ДЛЯ ВЫЯВЛЕНИЯ ОШИБОК ВЫПОЛНЕНИЯ УПРАЖНЕНИЙ НА SMART-ТРЕНАЖЕРЕ



**А.В. Кукареко**

Магистрант кафедры информатики БГУИР.



**С.Н. Нестеренков**

Доцент кафедры программного обеспечения информационных технологий, кандидат технических наук, доцент

Белорусский государственный университет информатики и радиоэлектроники  
E-mail: sasha.kukareko@gmail.com, nsn@bsuir.by

### **А.В. Кукареко**

Окончил Белорусский государственный университет информатики и радиоэлектроники в 2017. Магистрант БГУИР.

### **С.Н. Нестеренков**

Окончил БГУИР в 2007 году по специальности "Программное обеспечение информационных технологий", окончил магистратуру БГУИР в 2008 по специальности "Системный анализ, управление и обработка информации", окончил аспирантуру БГУИР в 2013 по специальности "Системный анализ, управление и обработка информации", окончил магистратуру БГУИР в 2013 по специальности "Экономика и управление народным хозяйством", в 2017 защитил диссертацию на соискание ученой степени кандидата технических наук по специальности "Системный анализ, управление и обработка информации".

**Аннотация.** В докладе рассматривается возможность использования алгоритмов машинного обучения и нейронных сетей для классификации качества выполнения упражнения по показаниям с датчика установленного на "умном" тренажере. Демонстрируется полученный результат и оценка решения задачи. Использовались следующие виды алгоритмов машинного обучения: метод опорных векторов, LSTM-сети и сверточные LSTM-сети.

**Ключевые слова:** Интернет вещей, временные ряды, нейронные сети, LSTM, сверточные LSTM, метод опорных векторов.

В настоящее время активно развиваются и повсеместно внедряются технологии «интернета вещей». Интернет Вещей (Internet of Things – IoT) – это концепция, в которой устройствам (объектам) реального мира, оснащенным различными датчиками и доступом в интернет, предоставлена возможность автономного обмена полезной информацией между собой, сервером и другими потребителями [1]. С активным внедрением технологий 5G, по прогнозам многих аналитиков, «вещей» (things) в сети появится в тысячи раз больше при гораздо большем трафике, скорости и мизерной задержке передачи данных. Одними из ключевых направлений развития и внедрения «интернета вещей» являются отрасли здравоохранения, реабилитации и фитнеса. Основными преимуществами использования интернета вещей и других smart-решений для сферы здравоохранения являются:

- повышенная мобильность сотрудников, продуктивность их работы;
- сбор данных о пациенте с последующим анализом;
- снижение риска ошибки и просчета из-за человеческого фактора;

- уменьшение затраты на лечение и восстановление;
- внедрение новых методов профилактики и диагностики заболеваний.

Своевременный анализ данных пациента может значительно сократить срок его реабилитации либо предупредить наступление заболевания. Активное внедрение «умных» датчиков в различные устройства, помогает и упрощает диагностику и анализ, однако возрастающее количество данных затрудняет анализ человеком в режиме реального времени. Для решения этой проблемы применяются различные алгоритмы, в том числе машинное обучение.

В статье рассматривается тренажер с 6-ти осевым датчиком положения (гироскоп и акселерометр). Датчик работает с частотой 20Hz и оснащен вычислительным модулем, который предоставляет еще 3 дополнительных параметра: pitch, roll, yaw. Пример выходных данных можно посмотреть в таблице 1.

Таблица 1. – Выходные данные 6-ти осевого датчика.

acc_x	acc_y	acc_z	gyro_x	gyro_y	gyro_z	roll	pitch	yaw
0.0418	-0.3922	-0.2517	-0.0609	2.0121	-0.1829	2.5111	-26.991	4.1909
0.0483	-0.3905	-0.2476	-0.1219	2.4390	-0.3048	2.5045	-26.877	4.1830
0.0456	-0.3877	-0.2473	-0.0609	2.8048	-0.3048	2.4871	-26.745	4.1745

Где:

- acc\_x, acc\_y, acc\_z – параметры акселерометра;
- gyro\_x, gyro\_y, gyro\_z – параметры гироскопа.
- pitch (тангаж) - угол вращения вокруг оси x относительно датчика в диапазоне от -180 до 180 градусов;
- roll (крен) - угол вращения вокруг оси y относительно датчика в диапазоне от -90 до 90 градусов;
- yaw (Рысканье) - угол вращения вокруг оси z относительно датчика в диапазоне от 0 до 360 градусов.

Параметры roll, pitch и yaw вычисляются датчиком в режиме реального времени, являются частью «связанной системы координат» и применяются во многих «смарт» устройствах, например в смартфонах (рисунок 1), предоставляя дополнительные возможности и данные для анализа.

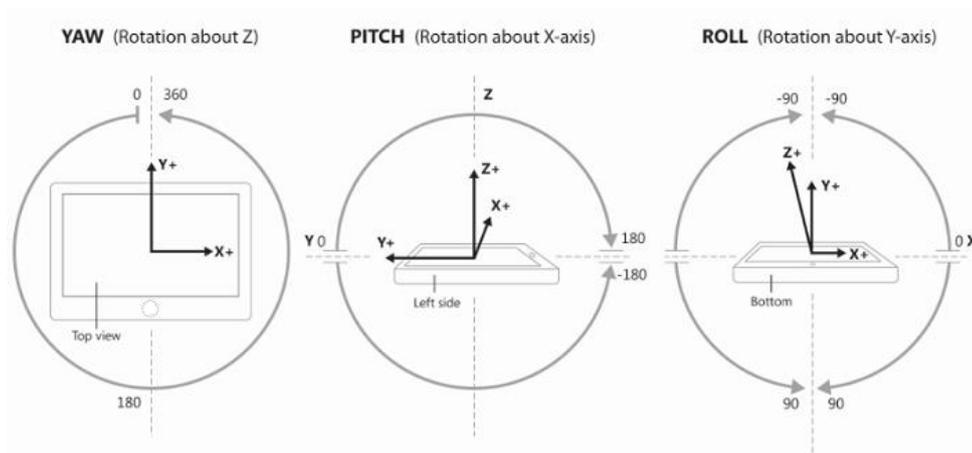


Рисунок 1. – Roll, pitch и yaw в «смарт» устройствах.

В статье рассматривается тренажер, в котором датчик установлен таким образом, чтобы вращение происходило относительно оси x. Соответственно параметр «pitch» является наиболее информативным.

Упражнения на тренажере представляют из себя последовательные маятникообразные движения. Задача пациента выполнять движения равномерно, соблюдая заданный системой или доктором интервал для повторения и диапазон (угол отклонения). На рисунке 2 представлен график «хорошо» выполненного упражнения.

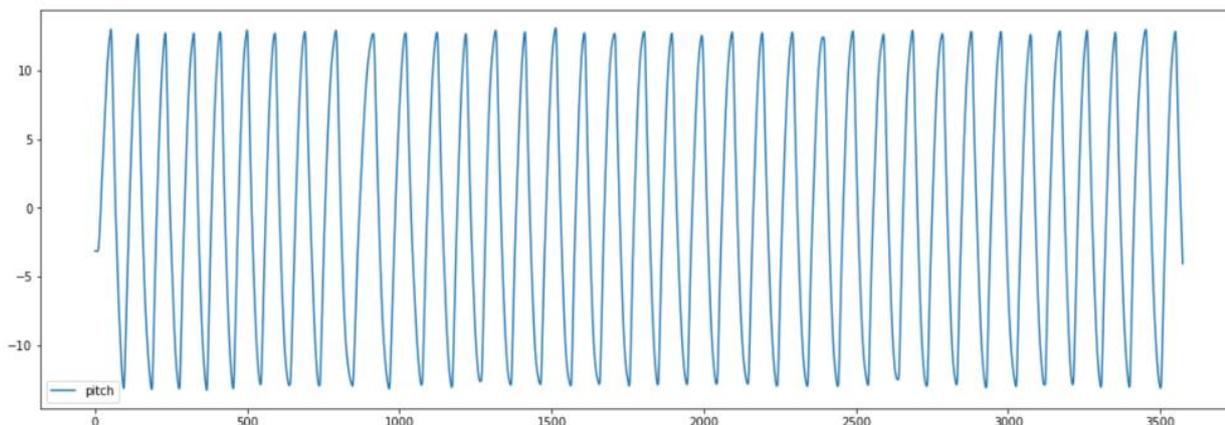


Рисунок 2. – График хорошо выполненного упражнения.

Отсутствие правильной техники и систематические ошибки во время выполнения упражнения могут не только негативно сказаться на скорости реабилитации, но также могут привести к серьезным травмам. Быстрое выявление таких ошибок и их исправление является неотъемлемой частью деятельности доктора, который проводит реабилитацию. Поэтому постоянное наблюдение доктора за пациентом и его техникой является важной стратегией реабилитации.

Ошибки во время выполнения упражнения могут быть как единичными, так и систематическими, могут появляться как в начале, середине так и в конце упражнения, примеры ошибок во время выполнения упражнения показаны на рисунке 3.

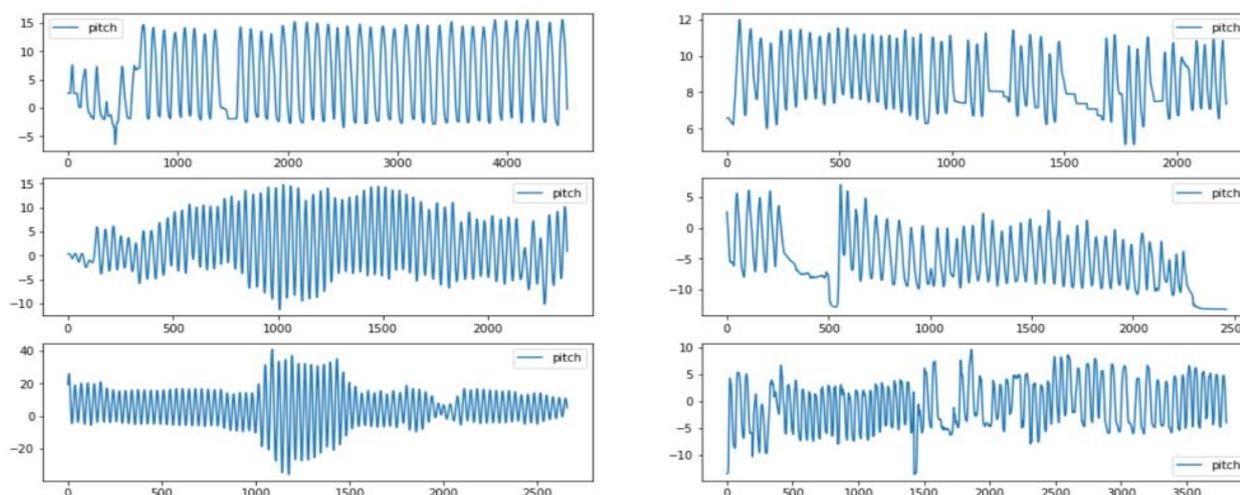


Рисунок 3. – Графики плохо выполненных упражнений.

Данные об упражнении представлены в виде временных рядов не фиксированной длины. Длина упражнения зависит от количества повторений и скорости их выполнения «пациентом». Выявление ошибок выполнения упражнения близкое к реальному времени поможет значительно улучшить технику выполнения упражнения «пациентом» и снизит риск ошибки из-за человеческого фактора.

Данную проблему можно свести к «классификации последовательностей» (sequence classification). Для решения такого рода задач часто применяются следующие алгоритмы [2].

Метод опорных векторов (SVM) - это набор алгоритмов обучения с учителем, использующихся, в основном, для задач классификации. Основная идея алгоритма состоит в том, чтобы построить гиперплоскость и разделить таким образом объекты выборки оптимальным способом. Алгоритм работает в предположении, что чем больше расстояние между разделяющей гиперплоскостью и объектами разделяемых классов, тем меньше будет средняя ошибка классификатора.

Преимуществами метода опорных векторов является:

- высокое быстродействие;
- единственно верное решение, за счет сведения решения к задаче выпуклого квадратичного программирования.

Недостатками метода опорных векторов являются:

- чувствительность к шумам;
- стандартизации исходных данных;
- отсутствие общего подхода к автоматическому выбору ядра в случае линейно неразделимости классов [3].

Так как данные упражнения являются набором последовательностей положения датчика, важным является не только анализ самих признаков, а так же анализ их последовательности. Для решения такого типа задач используют нейронные сети, способные "запоминать" контекст, а именно рекуррентные нейронные сети (RNN). В структуре рекуррентных нейронных сетей присутствуют механизмы, позволяющие запоминать информацию, и использовать её для решения последующих задач.

Долгая краткосрочная память (Long short-term memory; LSTM) – особая разновидность архитектуры рекуррентных нейронных сетей, способная к обучению долгосрочным зависимостям и на многих задачах значительно превосходит стандартную версию (RNN). LSTM-сети хорошо приспособлены к обучению на задачах классификации, обработки и временных рядов в случаях с неопределённой продолжительностью и границами.

LSTM слой обычно состоит из N модулей. Подробная структура «базового» модуля LSTM показана на рис 4. В модуль LSTM размещены три «гейта», а именно входной (input gate), выходной (output gate) и очистки памяти (forget gate). Эти три элемента играют ключевую роль для определения того, какую информацию следует запомнить. Временная память в LSTM реализована благодаря поочередному переключению «гейтов», это так же решает проблему вымывания и взрыва градиента, что актуально для RNN сетей.

Основой LSTM является состояние ячейки (cell state) - горизонтальная линия, проходящая сквозь верхнюю часть диаграммы.

Состояние ячейки - это что-то типа ленты конвейера. Она движется прямо вдоль всей цепи модулей только лишь с небольшими линейными взаимодействиями. Информация может просто проходить по ней без изменений.

LSTM имеет способность удалять или добавлять информацию из ячейки состояния, однако эта способность тщательно регулируется структурами, называемыми «гейтами» (gates).

Гейт - это способ избирательно пропускать информацию. Он состоит из sigmoid-слоя и операции поточечного умножения (pointwise multiplication).

На вход sigmoid-слоя поступает значение между 0 и 1, определяя тем самым пропускную способность гейта. Ноль - “ничего не пропускать”, один - “пропускать все”.

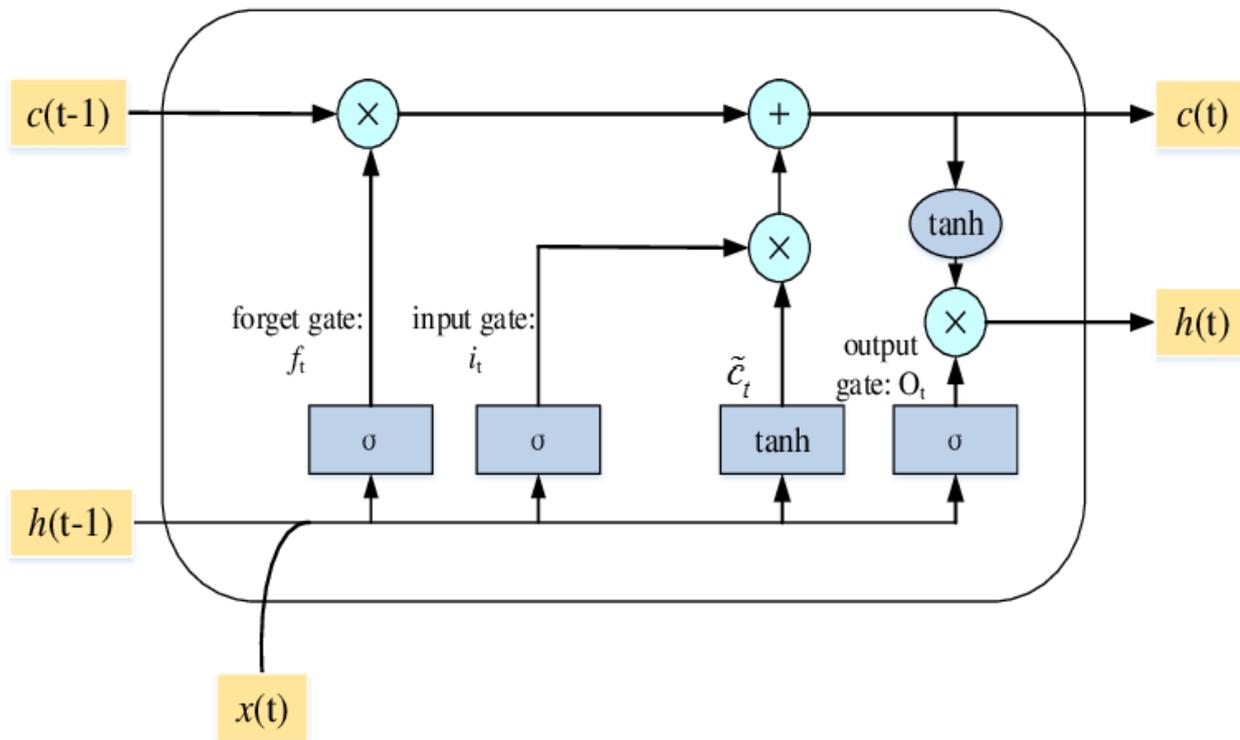


Рисунок 4. –Структура LSTM модуля.

Принцип работы модуля LSTM:

- На первом этапе модулю нужно решить, какую информацию нужно выбросить из состояния ячейки. Это решение принимается sigmoid-слоем, называемым «гейтом очистки памяти». Он получает на вход  $h(t-1)$  и  $x(t)$  и выдает на выходе число от 0 до 1 для каждого номера в состоянии ячейки  $c(t-1)$ . 1 означает «сохранить», а 0 — «избавиться»;
- На втором этапе нужно решить, какую информацию стоит сохранить в ячейку. Это действие проходит в два этапа. В первом этапе, входной sigmoid-слоем (input gate) решает, какие значения нужно обновить. Во втором этапе  $\tanh$ -слой создает новый вектор  $c(t)$  кандидатов на новые значения, который может быть сохранен в ячейку;
- Следующим этапом является обновление состояния ячейки  $c(t-1)$  до нового состояния  $c(t)$ : состояние ячейки на  $f(t)$  "гейт очистки памяти" (forget gate), «очищая» то, что ранее было решено «очистить». Затем прибавляем  $i(t) * c(t)$  – новые значения-кандидаты;
- В заключительном этапе нужно решить, какой результат мы собираемся подать на выход. Результат будет являться отфильтрованным состоянием ячейки. Сначала sigmoid-слоем решает, какие части состояния ячейки подать на выход. Затем состояние ячейки через пропускается через  $\tanh$ -слой, для преобразования сигнала в диапазон  $[-1, 1]$ , и затем умножаем его на выход sigmoid-слоя [4].

Так же в целях повышения качества классификации применяются комбинированные решения, например такие как гибридный сверточной нейронной сети (CNN) и LSTM-сети.

Сверточные нейронные сети (CNN - convolutional neural network) - специальная архитектура искусственных нейронных сетей. Название сеть получила из-за наличия операции «свертки». Данный тип сети получил широкое распространение после победы на конкурсе ImageNet. Изначально сеть использовалась для классификации изображений, затем

данный подход был применен и к другим типам задач [5]. Слои свёртки и субдискретизации состоят из нескольких «уровней» нейронов, называемых картами признаков (feature maps). Каждый нейрон такого слоя соединён с небольшим участком предыдущего слоя, называемым рецептивным полем. В случае изображения, карта признаков является двумерным массивом нейронов, или просто матрицей. Другие измерения могут быть использованы, если на вход принимается другой вид данных, например, аудио данные (одномерный массив) или объёмные данные (трёхмерный массив).

Операция свертки позволяет нейронной сети выполнять роль генератора новых или скрытых признаков из "сырых" данных, комбинируя сверточные слои можно на выходе получить более абстрактные признак. В то же время LSTM-сети включают в себя память для моделирования временных зависимостей в задачах временных рядов, и комбинация CNN и LSTM в одной архитектуре уже показала хорошие результаты в таких областях как распознавание речи, генерации описания по видео или изображению. Архитектура такого типа способна фиксировать временные зависимости от признаков, выделенных сверточными слоями.

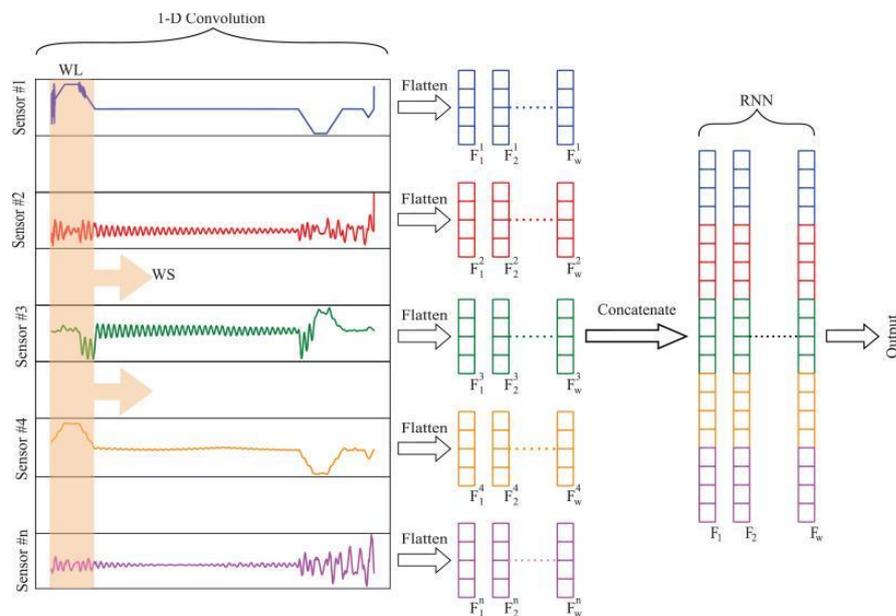


Рисунок 5. – Архитектура CNN LSTM сети.

Архитектура CNN LSTM-сети (рисунок 5) включает в себя одноуровневые слои сверточной нейронной сети (CNN), которые используются для выделения признаков, а затем новые признаки поступают в слои LSTM-сети, которые в свою очередь выявляют временные зависимости.

Для проведения экспериментов была проведена работа по подготовке данных:

– из базы данных упражнений было выделено 1523 файлов «хорошо» выполненных упражнений и 1569 файлов «плохо» выполненных упражнений;

– так как входной вектор модели должен быть фиксированным, а время выполнения упражнения является плавающей величиной, каждый файл был разбит на «окна» состоящие из 80 элементов с нахлестом в 50 %, что эквивалентно 4 секундам;

– датасет составил порядка 85997 «окон»;

– затем датасет был разделен на выборки train - 73360 записей и test - 12600 записей.

Обучающий набор содержит 36759 записи для "хорошо" выполненных упражнений и

"36600" записи для "плохо" выполненных упражнений (рисунок 6). Исходя из этого, можно сделать вывод, что классы сбалансированы, так как доля миноритарного класса составляет 99.5% от мажоритарного.

Так как классы являются сбалансированными, для определения точности модели можно использовать метрику «ассигасу» - «доля правильных ответов».

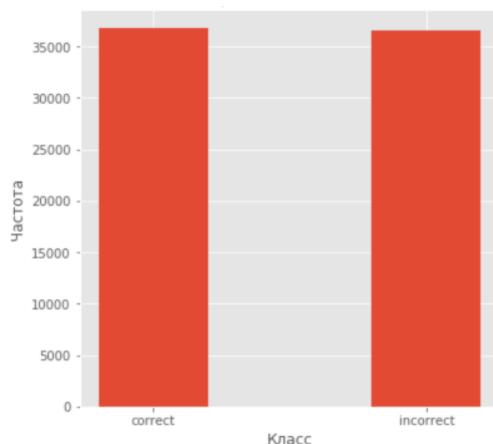


Рисунок 6. – Гистограмма распределения классов.

Первая модель построена на базе метода опорных векторов, с использованием библиотеки “scikit-learn”.

Модель обучалась со следующими параметрами:

- размер вектора признаков – 720;
- ядро – rbf;
- регуляризация C – 1.

Обучение модели заняло 24 минуты 21 секунду, результаты обучения можно увидеть в таблице 2.

Таблица 2. – Результаты обучения модели на базе метода опорных векторов.

Набор данных	Точность «ассигасу»
Train: 73360	0.915
Test: 12600	0.896

Следующая модель построена на базе LSTM-сети, с использованием библиотеки keras. На рисунке 7 изображена архитектура модели, она состоит из следующих слоев:

1. Input (входной слой): размерность – 80x9;
2. LSTM-слой: выходная размерность: 120, функция активации: гиперболический тангенс;
3. Dropout-слой: коэффициент сброса – 0.5;
4. Dense-слой: размерность: 100, функция активации: relu, и l2 регуляризацией: 0.01;
5. Dense-слой: размерность: 2, функция активации: softmax;
6. В качестве оптимизатора был использован Adam;
7. В качестве функции потерь была использована функция перекрестной энтропии;
8. Количество эпох: 10;
9. batch\_size: 64.

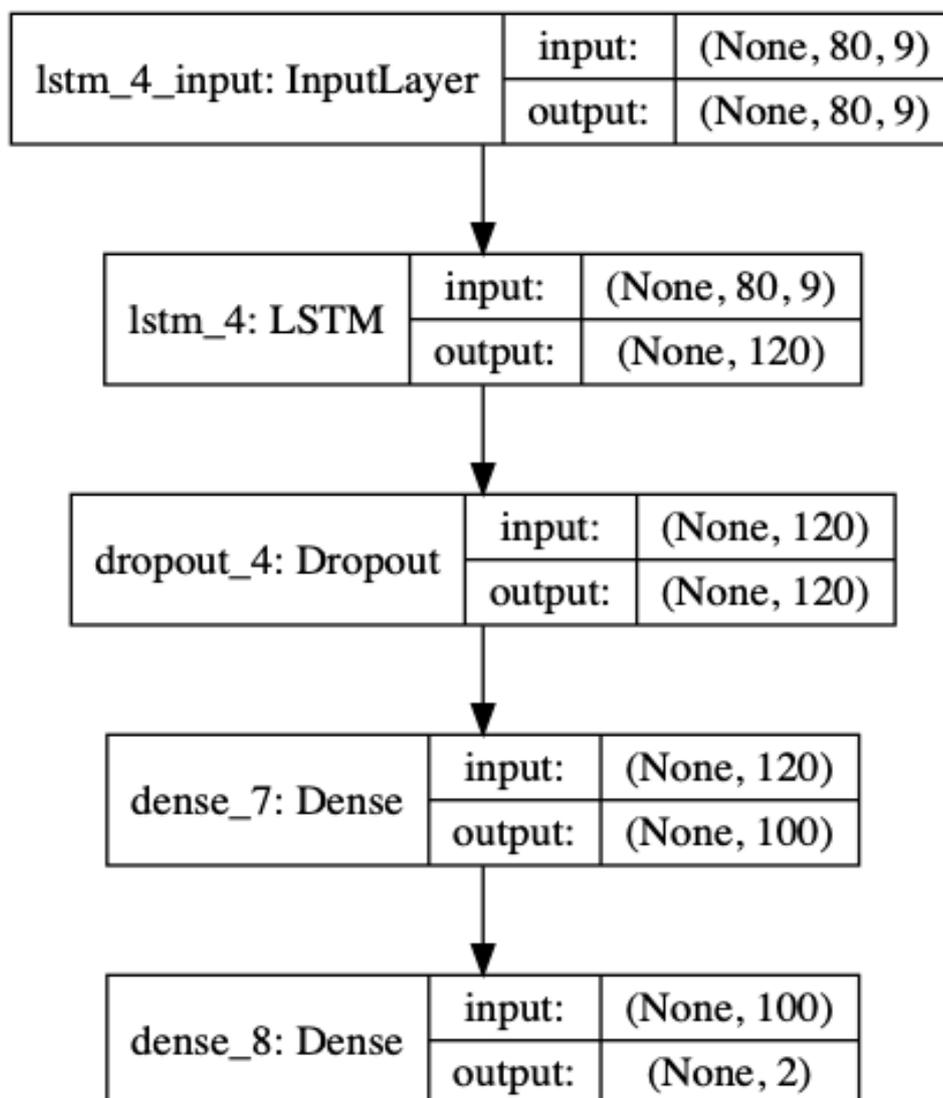


Рисунок 7. – Архитектура модели на базе LSTM-сети.

Обучение модели заняло 14 минут 46 секунд, результаты обучения можно увидеть в таблице 3.

Таблица 3. – Результаты обучения модели на базе LSTM-сети.

Набор данных	Точность «accuracy»
Train: 62355	0.9735
Valid: 11004	0.9608
Test: 12600	0.9475

Ниже на рисунке 8 представлены кривые обучения модели на базе LSTM на тренировочной и валидационной выборках.

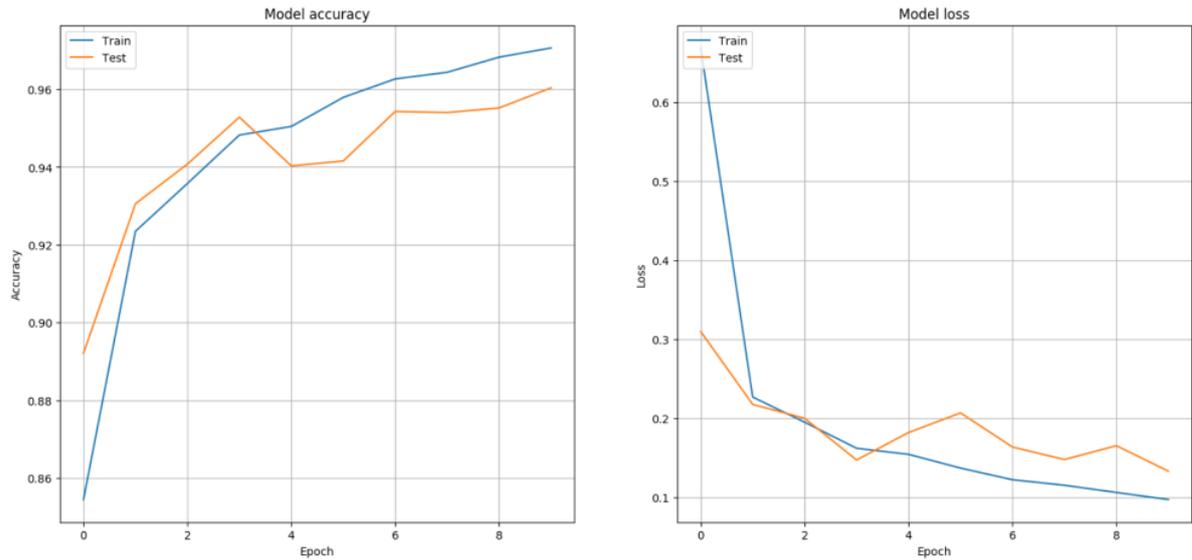


Рисунок 8. – Кривые обучения модели на базе LSTM-сети.

Финальная модель это гибрид CNN и LSTM-сети, построена с использованием библиотеки keras.

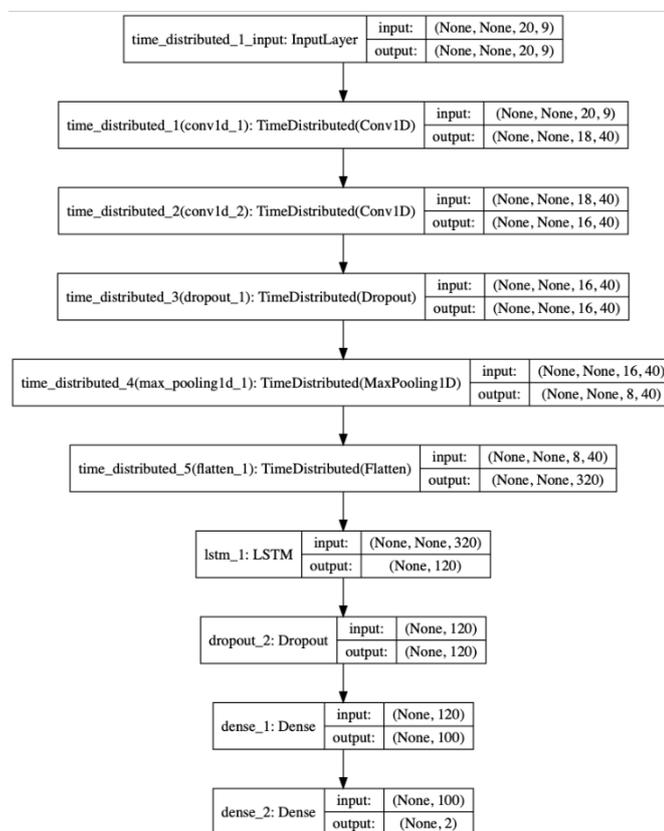


Рисунок 9. – Архитектура модели на базе CNN-LSTM-сети.

На рисунке 9 изображена архитектура модели, она состоит из следующих слоев:  
1. TimeDistributed входной-слой: размерность – 20x9;

2. TimeDistributed сверточный-слой: размерность: 1, количество фильтров: 40, функция активации: ReLU, размер ядра: 3;
3. TimeDistributed сверточный-слой: размерность: 1, количество фильтров: 40, функция активации: ReLU, размер ядра: 3;
4. TimeDistributed Dropout-слой: коэффициент сброса – 0.5;
5. TimeDistributed MaxPooling-слой: размерность: 1, размер пула: 2;
6. TimeDistributed Flatten-слой;
7. LSTM-слой: выходная размерность: 120, функция активации: гиперболический тангенс;
8. Dropout-слой: коэффициент сброса – 0.5;
9. Dense-слой: размерность: 100, функция активации: relu, и l2 регуляризацией: 0.05;
10. Dense-слой: размерность: 2, функция активации: softmax;
11. В качестве оптимизатора был использован Adam;
12. В качестве функции потерь была использована функция перекрестной энтропии;
13. Количество эпох: 10;
14. batch\_size: 64.

Обучение модели заняло 3 минуты 39 секунд, результаты обучения можно увидеть в таблице 4.

Таблица 4. – Результаты обучения модели на базе CNN-LSTM-сети.

Набор данных	Точность «accuracy»
Train: 62355	0.9715
Valid: 11004	0.9651
Test: 12600	0.955

На рисунке 10 представлены кривые обучения модели CNN-LSTM на тренировочной и валидационной выборках.

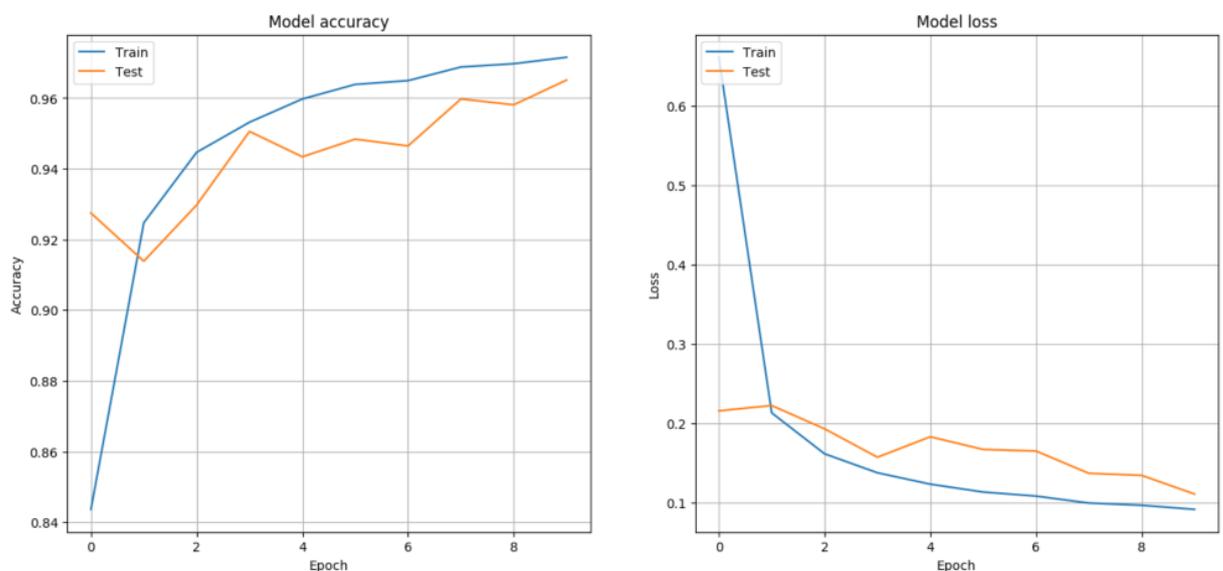


Рисунок 10. – Кривые обучения модели на базе CNN-LSTM-сети.

Таблица 5. – Сравнение результатов моделей.

Модель	Точность «ассигасу»	Время обучения
SVM	0.896	24:21
LSTM	0.9475	14:46
CNN-LSTM	0.955	3:39

В таблице 5 показана сравнительная характеристика моделей: точность и время за которое они обучились. Хуже всего показала себя модель построенная на базе метода опорных векторов, у неё самая низкая точность и самое большое время обучения. В то же время модели на базе LSTM показали значительно лучший результат за меньшее время обучения, что свидетельствует об эффективности этих моделей для решения поставленной задачи.

Гибридная модель CNN-LSTM показала самый лучший результат и самое маленькое время обучения.

Высокая точность обученной модели говорит о том, что её можно применять в «смарт» системах реабилитации для выявления ошибок выполнения упражнения, тем самым повысив эффективность реабилитации, снизив нагрузку на докторов и повысив удовлетворённость «пациентов».

#### Список литературы

[1.] Нестеренков, С. Н. Применения цифровых двойников и бессерверной архитектуры для управления IoT устройствами / С. Н. Нестеренков, А. В. Кукарёко // Информационные технологии и системы 2019 (ИТС 2019) : материалы междунар. науч. конф., Минск, 30 окт. 2019 г. / Белорус. гос. ун-т информатики и радиоэлектроники ; редкол.: Л. Ю. Шилин [и др.]. - Минск, 2019. - С. 70-71.

[2.] MEX: Multi-modal Exercises Dataset for Human Activity Recognition / A. Wijekoon, N. Wiratunga, K. Cooper [Electronic resource]. – Mode of access: <https://arxiv.org/pdf/1908.08992.pdf>. – Date of access: 24.12.2019.

[3.] Использование метода опорных векторов в задачах классификации / Баев Н.О. // Международный журнал информационных технологий и энергоэффективности. – 2017. – Т.2 №2(4) с. 17-21.

[4.] Как понять LSTM сети - Перевод статьи Кристофера Олаха "Understanding LSTM Networks" [Electronic resource]. – Mode of access: <https://alexsohn.github.io/ml/2015/11/17/LSTM.html> - Date of access: 05.01.2020.

[5.] Нестеренков, С. Н. Использование сверточных нейронных сетей для классификации и анализа тональности текстов / С. Н. Нестеренков, П. А. Федоров, В. А. Денисов // Информационные технологии и системы 2019 (ИТС 2019) : материалы междунар. науч. конф., Минск, 30 окт. 2019 г. / Белорус. гос. ун-т информатики и радиоэлектроники ; редкол.: Л. Ю. Шилин [и др.]. - Минск, 2019. - С. 248-249.

## USING MACHINE LEARNING TO IDENTIFY SERIOUS MISTAKES WHEN TRAINING ON THE SMART TRAINING MACHINE

*A.V.Kukareko*

*Master's Department  
Informatics BSUIR*

*S.N. Nesterenkov*

*PhD Associate professor of department of the  
software of information technologies*

*Belarusian State University of Informatics and Radioelectronics  
E-mail: sasha.kukareko@gmail.com, nsn@bsuir.by*

**Abstract.** The report considers the possibility of using machine learning algorithms and neural networks to classify the quality of the exercises according to data from a sensor installed on a smart training machine. The reached result is demonstrated along with the whole solution evaluation. The following types of machine learning algorithms were used: SVM, LSTM, and convolutional-LSTM networks.

**Keywords:** IoT, time series, classification, neural networks, LSTM, CNN-LSTM, SVM.