

## ПРИМЕНЕНИЕ НЕЙРОННОЙ СЕТИ ДЛЯ ПОЗИЦИОНИРОВАНИЯ МОБИЛЬНОГО ОБЪЕКТА ВНУТРИ ПОМЕЩЕНИЙ

*Е.В. Кокорева, к.т.н., доцент, Сибирский государственный университет телекоммуникаций и информатики, kokoreva@sibguti.ru.*

**УДК 004.896**

**Аннотация.** В современных условиях, связанных с цифровизацией общественной жизни и развитием инфокоммуникационных технологий, большой интерес вызывают сервисы, основанные на геолокации. Актуальной задачей является внедрение искусственного интеллекта в технологии позиционирования. Статья посвящена разработке программных модулей для определения местоположения объекта на языке программирования *Python* с использованием библиотек машинного обучения *TensorFlow* и *Keras*.

**Ключевые слова:** позиционирование; *Wi-Fi*; искусственный интеллект; машинное обучение; нейронная сеть; *Python*; *TensorFlow*; *Keras*.

## APPLICATION OF NEURAL NETWORK TO POSITION A MOBILE OBJECT INDOORS

*E.V. Kokoreva, Candidate of Technical Sciences, Associate Professor, Siberian State University of Telecommunications and Information Science.*

**Annotation.** In modern conditions, related to the digitalization of public life and the infocommunication technologies development, location-based services are of great interest. An urgent task is the implementation of artificial intelligence for positioning technologies. The article is dedicated to the development of software modules for object's location in the Python programming language using the TensorFlow and Keras machine learning libraries.

**Keywords:** positioning; *Wi-Fi*; artificial intelligence; machine learning; neural network; *Python*; *TensorFlow*; *Keras*.

### Введение

Системы локального позиционирования и навигации прочно вошли во все сферы общественной жизни. *LBS*-сервисы (англ. *Location-Based Service*), такие как ориентирование в помещениях со сложной конфигурацией, логистические маршруты, отслеживание грузов при транспортировке, а также защита зданий от взломов и несанкционированных проникновений, робототехника и пр. для своей реализации используют множество подходов [1, 2].

Известны системы определения местоположения в наземных сетях, основанные на технологиях *Wi-Fi*, *RFID*, *Bluetooth*, *UWB*, *ZigBee* и пр. [3, 4]. Из них наибольший интерес представляют сети стандарта *IEEE 802.11* благодаря широкому распространению беспроводных локальных сетей *WLAN* (англ. *Wireless Local Area Networks*) не только в производственных помещениях или различных учреждениях, но и в центрах развлечения и досуга.

Упомянутые выше технологии, в том числе и *Wi-Fi*, используют различные физические параметры, измерение которых позволяет определить координаты мобильного объекта. К таким параметрам относятся: *RSSI* – уровень мощности сигнала на приемном устройстве, *FTM* – точное измерение времени, *ToA* – время полета сигнала,  *AoA* – угол прибытия сигнала и др.) [3, 5-8].

На измерениях мощности, времени или угла основаны различные механизмы локации, среди наиболее известных выделяются методы трилатерации (мультилатерации), триангуляции (ангуляции), сравнения с образцом, по «ближайшему» и пр. [1, 9-12].

В последние годы большое значение приобрели методы определения местоположения с применением искусственного интеллекта, нейронных сетей и машинного обучения [13].

В данной статье рассматривается разработанное авторами программное обеспечение для исследования методов машинного обучения с целью позиционирования объекта внутри помещений в сети *Wi-Fi*.

### **Характеристика искусственного интеллекта, нейронных сетей и машинного обучения**

Искусственный интеллект (ИИ) – это компьютерная технология, способная имитировать поведение человека, которая служит для решения задач, обычно требующих вмешательства человеческого интеллекта [14, 15]. Области применения ИИ расширяются с каждым годом. Согласно последнему отчету *Harvard Business Review* основными задачами искусственного интеллекта в бизнесе и на производстве являются:

- выявление и предотвращение нарушений безопасности;
- устранение технических проблем пользователей;
- сокращение управления производством внедрением автоматизации;
- оптимизация ценообразования на основе предсказания предпочтений заказчиков и др.

Ключевыми концепциями в области ИИ являются машинное обучение и нейронные сети. Машинное обучение предоставляет возможность системе получать знания, выявлять закономерности и принимать решения без прямых инструкций, получаемых от разработчика. Нейронная сеть (искусственная нейронная сеть) является частью машинного обучения и представляет собой математическую модель, построенную по принципу организации и функционирования нейронов человеческого мозга [16, 17].

Можно определить ряд основных задач, решаемых с помощью алгоритмов машинного обучения. К ним относятся:

- классификация – определение категории объекта по его признакам.
- кластеризация – распределение объектов по категориям, если число этих категорий заранее неизвестно.
- регрессия – прогнозирование значения функции на основе выборки объектов с различными признаками.
- выявление аномалий – выделение из выборки объектов, отличающихся от стандартных видов.

Для простоты восприятия типы машинного обучения принято делить на три группы:

- обучение с учителем предполагает, что данные, подготовленные для анализа, однозначно содержат правильный ответ, поэтому машине остается определить объект с подходящим признаком или вычислить результат;
- обучение без учителя означает, как правило, что объем исходных данных слишком велик для того, чтобы их можно было разметить. Машина сама должна выявить закономерности, определить верное решение или отсортировать объекты по заранее неизвестным признакам;

- обучение с подкреплением заключается в том, что машина не просто анализирует данные, а предпринимает некоторые действия, которые могут иметь краткосрочные или долгосрочные последствия (положительные или отрицательные). Задача обучения заключается в том, чтобы свести к минимуму отрицательные отклики и оптимизировать действия машины с целью получения более высокой награды.

Искусственные нейронные сети состоят из нескольких слоев (рис. 1):

- входного слоя;
- одного или нескольких скрытых слоев;
- выходного слоя.

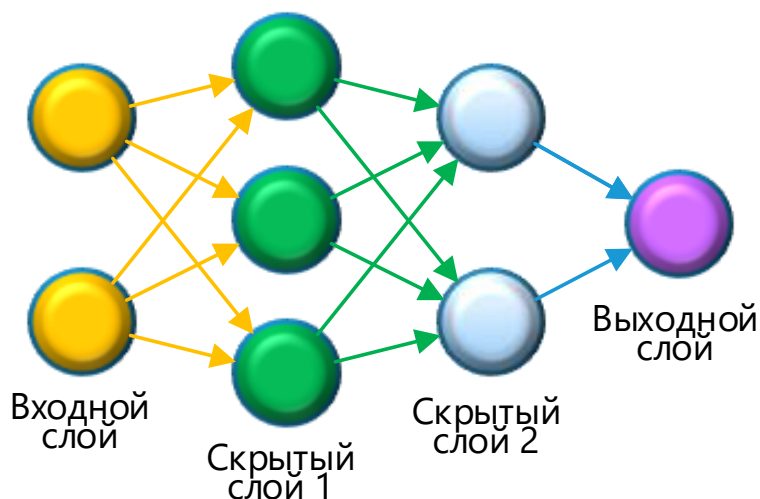


Рисунок 1

Каждый слой выполняет определенные функции в нейросети, например, группировки, обобщения, классификации или распознавания информации.

Прежде чем выполнить поставленную перед нейронной сетью задачу, она проходит многократно повторяющиеся этапы обучения на заранее подготовленных и размеченных наборах данных большого объема с установленными закономерностями. После чего обученная сеть может выявлять подобные закономерности на любых незнакомых ей наборах данных.

Имеется множество различных видов нейронных сетей, которые различаются по архитектуре, принципам машинного обучения и функционирования, но принято выделять три наиболее распространенных вида:

- нейронные сети прямого распространения *FFNN* (англ. *Feed Forward Neural Networks*). В них соседние узлы слоя не имеют связей друг с другом, а передача информации происходит напрямую от входного слоя к выходному;
- сверточные нейронные сети *CNN* (англ. *Convolutional Neural Network*). Они имеют многослойную структуру без обратных связей и нацелены в основном на решение задач по распознаванию образов;
- рекуррентные нейронные сети *RNN* (англ. *Recurrent Neural Network*) используют направленную последовательность связи между узлами. В таких сетях результат вычислений на каждом этапе используется в качестве исходных данных для следующего.

### Навигация мобильного объекта внутри помещений

Для построения системы определения местоположения на основе уже имеющейся в организации сети *Wi-Fi* была выбрана экспериментальная зона,

расположенная в коридорах 2-6 этажей здания. Были выявлены точки доступа AP (англ. *Access Point*), размещенные в различных помещениях учреждения и определены их координаты, после чего с помощью специально разработанного для этой цели программного обеспечения произведено большое количество измерений уровня мощности сигнала (*RSSI*), поступающего на мобильные устройства от двадцати семи обнаруженных точек доступа [18]. Измерения производились несколькими смартфонами в заранее размеченных образцовых точках коридоров здания через каждые 8 метров. Каждый смартфон выполнил в каждой образцовой точке по 10 замеров с интервалом 10 секунд.

Полученные результаты измерений были сведены в таблицы в виде, удобном для последующего машинного обучения. Фрагмент таких данных представлен в табл. 1.

Таблица 1.

№ Точки	AP1	AP2	AP3	...	AP25	AP26	AP27
305	-100	-61	-100	...	-100	-100	-89
305	-100	-60	-100	...	-96	-100	-86
305	-100	-60	-100	...	-98	-100	-88
305	-100	-61	-100	...	-96	-100	-87
306	-95	-62	-100	...	-100	-55	-100
306	-89	-65	-100	...	-100	-58	-100
306	-85	-65	-100	...	-100	-60	-100
306	-87	-64	-100	...	-100	-59	-100

Значение -100 дБм заменяет нулевые значения в ячейках, в случае, когда сигнал от AP не доступен в данной образцовой точке.

Подготовленные таким образом наборы измерений были использованы для обучения нейронной сети, а также в качестве тестовых выборок с целью определения местоположения мобильного объекта.

### **Применение специализированных библиотек языка Python**

Для машинного обучения применены специализированные библиотеки языка *Python*, такие как *TensorFlow* и *Keras* [19, 20].

*TensorFlow* – библиотека средств глубокого машинного обучения, разработанная компанией *Google* в виде надстройки к языку *Python*, в настоящее время может работать с другими языками программирования. *TensorFlow* объединяет в себе множество алгоритмов для создания нейронных сетей различного вида и назначения. Название библиотеки говорит о том, что она предназначена для эффективных вычислений с тензорами – многомерными массивами.

*Keras* – это API высокого уровня, который служит для облегчения процесса создания и обучения нейронной сети, основанной на применении средств библиотеки *TensorFlow*. *Keras* предлагает множество опций для работы с нейронной сетью, к ним относятся:

- функция активации – определяет выходное значение нейрона в зависимости от средневзвешенной суммы входных и порогового значений.
- оптимизаторы – содержат дополнительную информацию для повышения скорости и производительности конкретной модели.
- функция потерь – определяет потери из-за неправильного принятия решений на основе обучающих данных.
- метрика – используется для оценки качества работы конкретной модели.

Библиотека *Keras* содержит большое количество компонентов. Ниже рассматриваются наиболее часто используемые из них.

Функции активации:

- *sigmoid()* – сигмоидальная функция;
- *linear()* – линейная функция;
- *tanh()* – гиперболический тангенс;
- *relu()* – функция линейной ректификации;
- *exponential()* – экспоненциальная функция;
- *softmax()* – преобразует вектор значений в распределение вероятностей.

Оптимизаторы:

- *SGD()* – стохастический градиентный спуск;
- *RMSprop()* – алгоритм обратного распространения ошибки;
- *Adagrad()* использует алгоритм *Adagrad*, который адаптирует скорость обучения к частоте обновления параметров обучения, что ускоряет сходимость при сильно разреженных данных;
- *Adam()* использует алгоритм *Adam*, который является модификацией алгоритма *Adagrad*, использующей сглаженные версии среднего и среднеквадратического градиентов.

Функции потерь:

- *CategoricalCrossentropy()* – категориальная перекрестная энтропия;
- *BinaryCrossentropy()* – двоичная перекрестная энтропия;
- *MeanSquaredError()* – среднеквадратическая ошибка;
- *KLDivergence()* – расстояние Кульбака — Лейблера;
- *CosineSimilarity()* – косинус угла между реальным и предсказанным значением.

Метрики:

- *Accuracy()* – количество положительных исходов обучения;
- *BinaryAccuracy()* – бинарная точность;
- *AUC()* – площадь под кривой ошибок. Чем ближе значение этого показателя к единице, тем выше качество модели;
- *Precision()* – количество истинно положительных исходов обучения из всех исходов, которые были определены как положительные;
- *Recall()* – полнота, или чувствительность, показывает, какую долю положительных исходов удалось выявить.

### Моделирование на языке *Python*

После проведения некоторого количества экспериментов для дальнейшего исследования из предложенных функций активации были выбраны гиперболический тангенс и функция линейной ректификации. В качестве оптимизатора выбран метод обратного распространения ошибки, а в качестве функции потерь – среднеквадратическая ошибка и определена скорость обучения – 0,0001 (значения большего или меньшего порядка не дали лучших результатов). Были опробованы модели нейронной сети с одним, тремя и пятью слоями и метод машинного обучения – с учителем.

Создание однослойной модели:

```
model = tf.keras.Sequential([keras.layers.Flatten(input_shape = [27, ]),
```

```
Dense(units = 1, activation = 'relu')])
```

Здесь *input\_shape* – это размер входного вектора, *units* – количество нейронов в слое.

Создание трехслойной модели:

```
model = tf.keras.Sequential([keras.layers.Flatten(input_shape = [27, 1]),  
                             Dense(units = 200, activation = 'relu'),  
                             Dense(units = 50, activation = 'relu'),  
                             Dense(units = 10, activation = 'relu')])
```

Компиляция модели:

```
model.compile(optimizer = keras.optimizers.RMSprop(  
              learning_rate = 0.0001,  
              centered = True),  
              loss = keras.losses.MeanSquaredError(),  
              metrics = [keras.metrics.Accuracy()])
```

Обучение модели:

```
model.fit(xs, ys, epochs = 2000)
```

Здесь *xs* – входная (обучающая) выборка, *ys* – вектор выходных значений (результат), *epochs* – количество эпох обучения.

Сохранение обученной нейронной сети:

```
file = 'model.keras'  
model.save(file)
```

Обученную нейронную сеть можно применить к тестовому набору данных для целей позиционирования мобильного объекта.

Предсказание результата с тестовыми данными из массива *p*:

```
yy = model.predict(p)
```

Результаты машинного обучения в виде вероятностей корректного определения местоположения объекта, погрешности не более восьми метров (в соседней точке) и более восьми метров приведены на рис. 2-4. Заметим, что в приведенном примере позиционирование осуществлялось на втором этаже здания, где полностью отсутствуют точки доступа, а сигнал проникает только с верхних этажей.

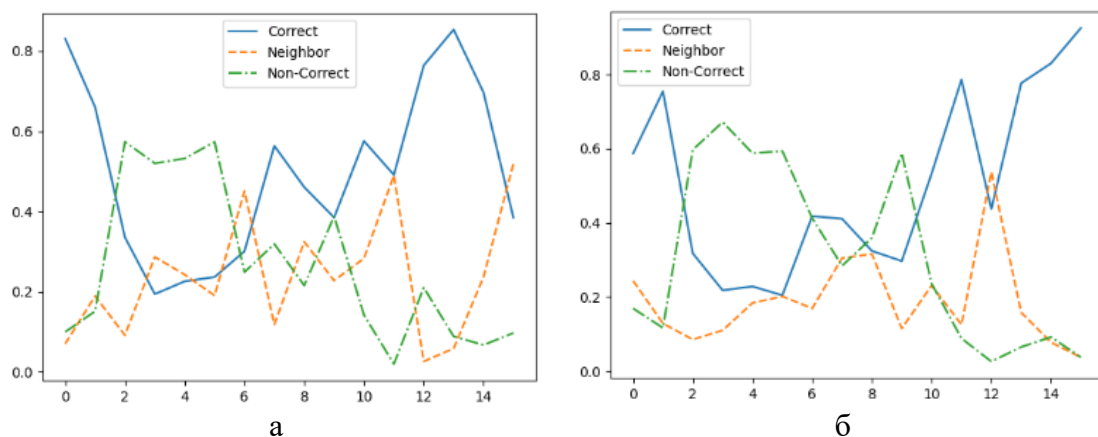


Рисунок 2

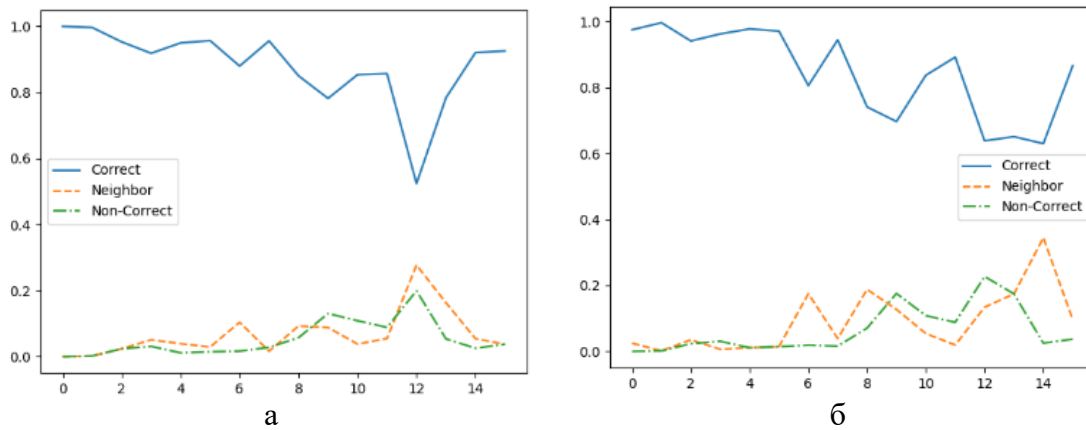


Рисунок 3

Рис. 2-4 содержат сравнение результатов машинного обучения с разным количеством слоев модели и двух функций активации '*relu*' и '*tanh*'. По оси абсцисс отложены условные номера образцовых точек, а по оси ординат – вероятности точного определения местоположения объекта (синий), с ошибкой не более 8 м. (оранжевый) и с погрешностью более 8 м. (зеленый).

На рис. 2 приведены, соответственно, характеристики однослойной нейронной сети: 2а – с функцией активации '*relu*'; 2б – для модели на основе функции '*tanh*'. Рис. 3а-б представляет те же характеристики, только для трехслойной модели, а рис. 4а-б – для пятислойной.

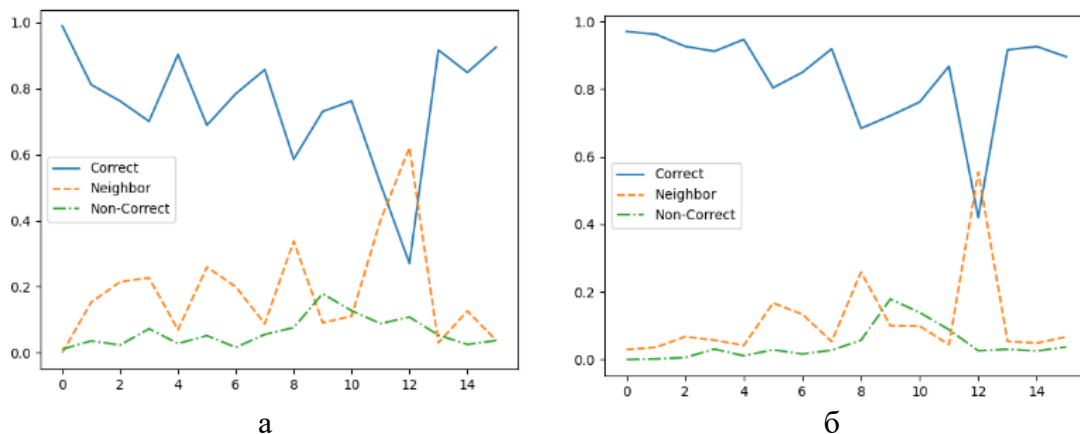


Рисунок 4

Можно видеть, что все однослойные нейронные сети показывают низкие по качеству результаты независимо от функции активации и скорости обучения. Вероятность ошибки в сумме по всем точкам для функции линейной ректификации составила 0,23, а для функции гиперболического тангенса – 0,27.

Рис. 2 показывает также, что в точках со 2-й по 6-ю в однослойной модели почти отсутствуют корректные и даже «соседние» результаты, поскольку мобильные устройства абонентов доступны для радиосигнала в этих местах только от одной или двух *AP*.

Хороший результат даже в плохих условиях радиопокрытия показывает нейронная сеть с функцией активации '*relu*'. Суммарная по всем точкам вероятность ошибки более 8 метров в 3-слойной нейронной сети имеет значение 0,03, а в 5-слойной – 0,05.

Функция гиперболического тангенса тоже может рассматриваться в качестве функции активации машинного обучения для определения местоположения

мобильного объекта, поскольку она дает немногим худшие результаты, чем функция линейной ректификации, а в случае пятислойной модели даже лучшие (вероятность ошибки 0,03).

### **Заключение**

Следует отметить, что представленные результаты для второго этажа здания иллюстрируют процесс определения местоположения с помощью методов машинного обучения в наихудших условиях, поскольку там полностью отсутствуют точки доступа, и сигнал поступает к устройству абонента от удаленных AP. Следовательно, улучшение радиопокрытия помещений приведет к улучшению результатов позиционирования.

Из приведенных выше исследований можно сделать вывод, что изученный подход к машинному обучению для целей определения местоположения мобильного объекта может быть применен при разработке системы локального позиционирования в сети *Wi-Fi*, поскольку правильный выбор опций нейронной сети может привести к высокоточным результатам (с наименьшей погрешностью).

Увеличивать количество слоев нейронной сети более трех не имеет смысла, поскольку точности вычислениям это не прибавляет, зато сильно увеличивает длительность машинного обучения.

Следующий этап работ в данной области предполагает исследование применения данных алгоритмов машинного обучения не только в сети *Wi-Fi*, но и в других применимых к позиционированию мобильного объекта технологиях для дальнейшего повышения точности.

### **Благодарности**

Исследования проводились в рамках государственного задания № 071-03-2024-008 «Прикладные научные исследования в области создания и развития перспективных средств и систем связи, устройств интеллектуальной электроники, инструментария высокопроизводительной обработки и защиты информации с внедрением результатов в образовательный процесс подготовки кадров для цифровой экономики (этап 1)» от 19.01.2024.

### **Литература**

1. Bensky A. *Wireless Positioning Technologies and Applications*, 2nd Edition. – Boston, London: Artech House, 2016. – 424 p.
2. Kolodziej K.W., Heilm J. *Local positioning systems : LBS applications and services*. – Boca Raton, FL: CRC/Taylor & Francis, 2006. – 488 p.
3. Zekavat R., Buehrer R. M. *Handbook of Position Location: Theory, Practice and Advances*. – Hoboken, New Jersey : John Wiley & Sons, Inc., 2012. – 1223 p.
4. *Wi-Fi location-based services 4.1 design guide*. – San Jose : Americas Headquarters Cisco Systems, Inc., 2008. – 206 p.
5. Кокорева Е.В., Костюкович А.Е. Результаты натурных испытаний системы определения местоположения в сети Wi-Fi // Экономика и качество систем связи, 2021. – № 3 (21). – С. 64-71.
6. Guoquan L., Enxu G., Zhouyang Y., Yongjun X., Jinzhao L., Yu P. Indoor positioning algorithm based on the improved RSSI distance model // *Sensors*, 2018. – V. 18 (9). – 15 p.
7. Bullmann M., Fetzer T., Ebner F., Ebner M., Deinzer F., Grzegorzec M. Comparison of 2.4 GHz WiFi FTM-And RSSI-based indoor positioning methods in realistic scenarios // *Sensors*, 2020. – V. 20 (16). – 26 p.



8. Banin L., Bar-Shalom O., Dvorecki N., Amizur Y. High-Accuracy Indoor Geolocation using Collaborative Time of Arrival [Электронный ресурс] – URL: [https://www.researchgate.net/publication/320146822\\_High-Accuracy\\_Indoor\\_Geolocation\\_using\\_Collaborative\\_Time\\_of\\_Arrival\\_CToA](https://www.researchgate.net/publication/320146822_High-Accuracy_Indoor_Geolocation_using_Collaborative_Time_of_Arrival_CToA) (дата обращения – октябрь 2024 г.).
9. Pradhan S., Bae Y., Pyun J.-Y., Ko N.Y., Hwang S.-s. Hybrid TOA trilateration algorithm based on line intersection and comparison approach of intersection distances // Energies, 2019. – V. 12 (9). – 26 p.
10. Cell Phone Trilateration Algorithm // 101 Computing [Электронный ресурс] – URL: <https://www.101computing.net/cell-phone-trilateration-algorithm/> (дата обращения – октябрь 2024 г.).
11. Tonello A.M., Inserra D. Radio positioning based on DoA estimation: an implementation perspective // IEEE International Conference on Communications 2013: IEEE ICC 2013-Workshop on Advances in Network Localization and Navigation (ANLN), 2013. – pp. 27-31.
12. Малодушев С.В., Рогов А.А. Определение локации в корпоративных Wi-Fi сетях // Вестник ЮУрГУ. Серия «Математическое моделирование и программирование», 2016. – Т. 9. – № 1. – С. 92-104.
13. Saleem F., Wyne S. Wlan-Based Indoor Localization Using Neural Networks // Journal of Electrical Engineering, 2016. – V. 67 (4). – pp. 299-306.
14. What Is AI? Learn About Artificial Intelligence [Электронный ресурс] – URL: <https://www.oracle.com/artificial-intelligence/what-is-ai/> (дата обращения – октябрь 2024 г.).
15. Hardesty L. Explained: Neural networks // MIT News [Электронный ресурс] – URL: <https://news.mit.edu/2017/explained-neural-networks-deep-learning-0414>. (дата обращения – октябрь 2024 г.).
16. Schmidhuber J. Deep Learning in Neural Networks: An Overview // Neural Networks, 2014. – V. 61. – pp. 85-117.
17. Джоши П. Искусственный интеллект с примерами на Python.: Пер. с англ. – СПб. : ООО «Диалектика», 2019. – 448 с.
18. Свидетельство о государственной регистрации программы для ЭВМ № 2021611439 Российская Федерация. Программа измерения уровня сигнала, принимаемого абонентским устройством в сети Wi-Fi, для определения местоположения абонента // Шурыгина К. И., Кокорева Е. В. Заявка № 2021610057 от 11.01.2021.
19. Gulli A., Karoor A., Pal S. Deep Learning with TensorFlow 2 and Keras, 2nd Edition. – Birmingham, Mumbai : Packt Publishing Ltd., 2019. – 647 p.
20. Жером О. Прикладное машинное обучение с помощью Scikit-Learn, Keras и TensorFlow, 2-е изд.: Пер с англ. – СПб. : ООО «Диалектика», 2020. – 1040 с.