

# Разработка системы проактивной локализации киберфизического пространства на основе методов машинного обучения\*

Д.А. Малов, А.Ю. Едемский, А.И. Савельев

Федеральное государственное бюджетное учреждение науки Санкт-Петербургский институт информатики и автоматизации РАН (СПИИРАН), г. Санкт-Петербург, Россия

**Аннотация.** Киберфизические системы (КФС) основаны на бесшовной интеграции вычислительных мощностей в физическую среду в организации, на предприятии или производстве. В данной работе предлагается система проактивной локализации для отслеживания и прогнозирования местоположения пользователей и мобильных роботов. Разработанная система позволяет предсказывать активность отслеживаемого объекта при помощи различных методов машинного обучения. В статье приведено сравнение моделей машинного обучения для системы локализации и навигации, а также представлена концепция системы проактивной локализации.

**Ключевые слова:** Киберфизическая система, прогнозирование временных рядов, проактивная локализация, рекуррентные нейронные сети, обучение с подкреплением.

DOI 10.14357/20718632180408

## Введение

Активное внедрение систем Интернета вещей в разные сферы современной жизни способствует созданию полностью децентрализованного социокиберфизического пространства, которое представляет собой интеграцию цифровых и физических компонентов в социальные структуры [1]. Физические компоненты данной системы могут быть представлены автономными роботизированными единицами и предоставлять пользователю разнообразные сервисы, такие как: навигация, корпоративное телевидение, видеоконференцсвязь, а также поиск и выдача справочной информации [2]. Поскольку

количество роботизированных единиц ограничено, необходимо разработать новый более эффективный способ их размещения в обслуживаемом пространстве, а также алгоритмы периодического изменения их координат на основе динамики изменения наиболее посещаемых зон. Поведение робота, основанное на текущем и будущем местоположениях пользователей, позволит оптимизировать ресурсы интеллектуального пространства.

Основная проблема проактивной системы локализации может быть формально определена как прогнозирование многомерных временных рядов, представляющие собой активность объектов в интеллектуальном децентрализованном пространстве [3-5]. Это может отно-

\*Исследование выполнено при финансовой поддержке Совета по грантам Президента Российской Федерации (проект № МК-383.2018.9)

ситься и к социокриберфизической системе с децентрализованными связями между участниками [6-8].

## 1. Обзор современных исследований в области локализации

В последнее время активное развитие получило направление исследований, связанное с прогнозированием местоположения объекта и активности человека в интеллектуальном пространстве. В [9, 10] рассмотрены возможные методы анализа временных рядов и предсказания локализации объектов для применения данной концепции в автотранспортных беспроводных децентрализованных самоорганизующихся (Vehicular ad hoc networks, VANETs) и мобильных сетях. Авторами этих работ проведен сравнительный анализ ряда методов предсказания местоположения, которые могут быть использованы транспортными средствами для оценки своего будущего местоположения, включая: счисление координат, нейронные сети, регрессия опорных векторов, фильтр Калмана и многочастичный фильтр. Результаты исследования показали, что такие методы, как счисление координат, фильтр Калмана и многочастичный фильтр, имеют наилучшие вычислительные характеристики, в частности, время отклика, тогда как методы машинного обучения обладают низкой эффективностью прогнозирования. Многочастичный фильтр и фильтр Калмана превосходят метод счисления координат по мере увеличения ошибки локализации, так как гауссовские ошибки могут влиять на линейный характер траектории движения транспортных средств. Основной причиной меньшей точности алгоритмов машинного обучения является «близорукость» для образцов локализации, поскольку данные методы опираются на изучение базовой структуры решаемой задачи и на поиск закономерностей по предоставленным данным.

В [11] авторы предлагают подход к определению местоположения объектов, называемый новым методом локализации (Novel Localization Method, LNM), который использует относительную мощность принимаемого Wi-Fi-

сигнала соседнего объекта для построения базы данных «отпечатков» пользования связью и применяет модель прогнозирования цепи Маркова для облегчения процесса позиционирования. Результаты исследования показали, что LNM позволяет определять местоположение объектов в реальном времени с высокой точностью. Кроме того, данный подход достиг уровня зрелости, который позволяет реализовать решения задачи локализации на практике и имеет потенциал для крупномасштабного внедрения в киберфизические системы.

Авторы работ [12-14] анализируют методы локализации и их иерархической таксономии, а также возможность применения их в различных системах. В основе таксономии методов локализации лежит наличие «офлайн»-обучения, а именно, самоопределения и обучения зависимым походам. Кроме того, в этих работах проведен обзор нерешенных вопросов в области схем локализации для Интернета вещей и предложен ряд направлений для дальнейших исследований.

В работах [15-18] авторами проанализированы возможные методы и алгоритмы для обеспечения работы «умных домов», в основе которых лежит анализ активности пользователей и возможность делать предсказания на основе имеющихся гетерогенных данных, что позволяет повысить степень автоматизации среды. Было проведено исследование не только методов прогнозирования местоположения и движения, но и геологического прогнозирования, которое может найти широкое применение в геолокационных сервисах и приложениях, что позволит значительно расширить сферу использования данной технологии.

Авторы работ [19,20] представляют подход к прогнозированию будущей активности в многопользовательском интеллектуальном пространстве с использованием метода встраивания, аналогичного алгоритму векторного представления слов в обработке естественного языка. Кроме подобного типа векторизации авторы предлагают использование нейронных сетей с долгой краткосрочной памятью (Long Short-Term Memory, LSTM) с целью прогнозирования одной векторной координаты для следующего движения на основании предыдущих

перемещений всей группы объектов. Затем по этому предсказанию производится отбор нескольких кандидатов для выполнения следующего действия посредством выбора векторов в окрестности выходного значения LSTM-сети. После интеграции действий в векторное пространство определялись семантически значимые связи между результирующими векторами. Авторы протестировали представленный подход, используя реальные данные, полученные при помощи многопользовательского испытательного стенда. Достоверность прогноза следующего действия составила приблизительно 82%. Предложенный метод может быть использован в многопользовательских интеллектуальных пространствах для разработки умных сервисов обслуживания, например, рекомендательных систем. Авторы работ акцентировали внимание на практической значимости глубоких LSTM нейронных сетей и сетей Wi-Fi в прогнозировании активности пользователей интеллектуального пространства и предложили заменить сенсорную систему моделями временных рядов, позволяющими прогнозировать количество пользователей в заданном месте в определенном временном интервале.

## 2. Алгоритмы анализа временных рядов

В [21] представлен обзор основных алгоритмов анализа временных рядов:

*А. Авторегрессионное интегрированное скользящее среднее (autoregressive integrated moving average, ARIMA)*

Алгоритм ARIMA также называют методом Бокса-Дженкинса. Компоненты дифференцирования, авторегрессии и скользящего среднего составляют нестационарную модель ARIMA, которую можно записать в виде линейной функции:

$$Y_t = c + \phi_1 y_{t-1} + \dots + \phi_p y_{t-p} + \dots + \theta_1 e_{t-1} + \theta_q e_{t-q} + e_t. \quad (1)$$

где  $c$  – константа,  $\phi_1, \phi_p$  – параметры модели,  $e_t$  – параметр ошибки,  $q$  – количество параметров для включения в модель,  $p$  – количество запаздываний, использованных в модели,  $\theta_1, \theta_q$  – параметры компоненты скользящего среднего.

*В. Деревья решений (Decision Tree) и случайный лес (Random Forest)*

Дерево решений является непараметрическим методом контролируемого обучения, используемого для решения задач классификации и регрессии. Проактивная система локализации должна уметь предсказывать координаты  $(x_i, y_i, z_i)$  следующей позиции пользователей внутри помещений, где развернута КФС. Таким образом, прогностический алгоритм должен предсказывать несколько вещественных чисел.

Random Forest (Случайный лес) – это ансамбль деревьев решений, который может быть использован для решения задач классификации и регрессии с несколькими выходными переменными. Само дерево решений представляет собой эффективный базовый классификатор для создания мультимножеств (бэггинг, bagging). Бэггинг является одним из первых простых типов ансамблей и основан на методе статистического бутстрэпа (bootstrap), который позволяет оценить большой объем статистических данных совместных распределений. Метод бутстрэпа реализуется следующим образом (Рис. 1): пусть  $X$  – выборочная совокупность размером  $N$ , при помощи которой сформируется новое множество  $X_1$  путем выбора образцов объекта из исходного набора  $N$  раз, предполагая, что события выбора равновероятны ( $P=1/N$ ). Стоит отметить, что по причине возврата объектов в оригинальную выборку перед формированием следующего подмножества, объекты будут повторяться. Таким образом при повторении данной процедуры  $M$  раз будут получены  $M$  выборочных подмножеств  $X_1, \dots, X_M$ . Наличие достаточно большого количества выборочных подмножеств позволят провести статистическую оценку исходного распределения.

Приведем краткое описание алгоритма бэггинга. Пусть  $X$  – обучающая выборка, из которой при помощи бутстрэп-алгоритма были получены подмножества  $X_1, \dots, X_M$  для последующего обучения классификатора  $clf_i(x)$ . Конечный классификатор усредняет прогнозы всех алгоритмов (в случае задачи классификации этот шаг соответствует голосованию):

$$clf(x) = \frac{1}{M} \sum_{i=1}^M clf_i(x) \quad (2)$$

Схема работы бэггинга показана на Рис. 2.

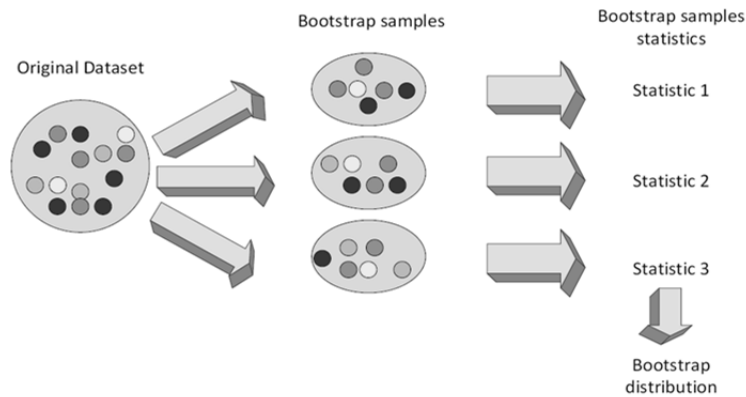


Рис. 1. Схема статистического бутстрэпа

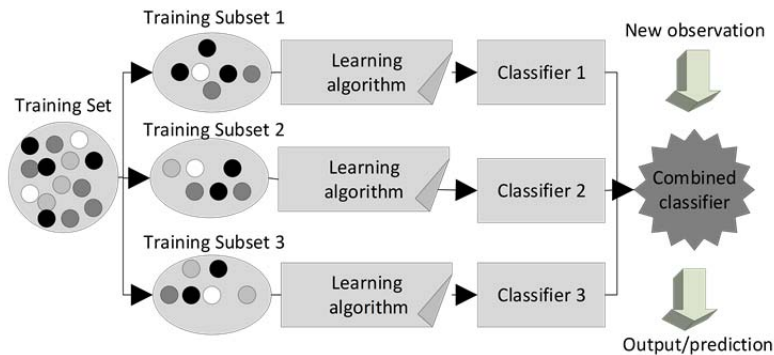


Рис. 2. Бэггинг-алгоритм

Для обучения разработанной модели были использованы данные, полученные посредством имитации перемещений пользователя в помещениях организации в среде Unity3D (Рис. 3). Эти данные содержат информацию о местонахождении пользователя в течение рабочего дня на протяжении примерно двух месяцев (23686 регистраций с датчиков присутствия) и могут быть представлены в виде набора данных со следующими параметрами:

- `datetime` - дата и время регистрации местоположения (например, 25/3/2018 11:16:25 AM);
- `user_id` - идентификатор пользователя, местоположение которого необходимо определить;
- `x_coordinate` – значение x-координаты пользователя;
- `z_coordinate` – значение z-координаты пользователя;
- `weekday` – день недели, представленный целым числом от 1 до 7;
- `time_in_seconds` - время, прошедшее от начала рабочего дня.

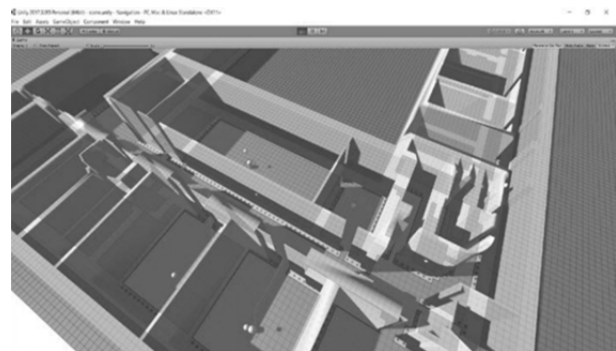


Рис. 3. Интерфейс симуляции перемещений пользователя в среде Unity3D

В качестве метрики использовалась абсолютная ошибка прогнозирования местоположения, выраженная в метрах:

$$error = \frac{1}{n} \sqrt{(x_{true} - x_{predicted})^2 + (z_{true} - z_{predicted})^2} \quad (3)$$

где  $x_{true}$ ,  $z_{true}$  и  $x_{predicted}$ ,  $z_{predicted}$  являются массивами истинных и прогнозируемых значений координат местоположения пользователя соответственно.

В качестве параметров процесса обучения были выбраны функции «weekday» и «time\_in\_seconds». Поскольку использовался метод перекрестной проверки на достоверность, полученная ошибка была рассчитана как среднее значение всех ошибок, заданных в подмножествах данных, проверенных перекрестно. Таким образом, для этих параметров ошибка предсказания координаты пользователя составляла приблизительно 9 метров для дерева решений и 10 метров для алгоритма случайного леса.

Очевидно, что такая ошибка неприемлемо высока для навигации внутри помещений, поэтому было решено добавить дополнительные параметры:

- `x_previous` - последнее зарегистрированное значение x-координаты пользователя;
- `z_previous` - последнее зарегистрированное значение z-координаты пользователя;
- `time_difference` - разница во времени между текущим и предыдущим показанием.

Для предложенных параметров ошибка прогнозирования координат пользователя составила приблизительно 6 метров для дерева решений и 0,4 метра для алгоритма случайного леса.

Далее была проведена еще одна симуляция в среде Unity3D, но с участием двух пользователей. После обучения модели средняя ошибка в прогнозировании координат пользователя со-

ставила в среднем 0,7 метра. В Табл. 1 показаны результаты обучения с различными параметрами.

Алгоритм случайный лес показал наименьшую ошибку при использовании обучающей выборки, равную 0,4 м. В следующих параграфах представлены результаты тестирования LSTM нейронных сетей и алгоритмов глубокого обучения с подкреплением для прогнозирования местоположения пользователей в интеллектуальном пространстве.

### С. LSTM и GRU нейронные сети

Рекуррентные нейронные сети (Recurrent Neural Networks, RNN), в отличие от нейронных сетей прямого распространения, способны использовать свое внутреннее состояние (память) для анализа последовательностей выходных данных. LSTM нейронные сети являются разновидностью рекуррентных нейронных сетей, которые активно применяются при решении регрессионных задач.

Скрытый слой рекуррентного блока (Gated recurrent unit, GRU) работает по тому же принципу, что и LSTM-сети, но имеет более упорядоченную структуру и, как следствие, требует меньше вычислительных мощностей при запуске. В то же время слои GRU могут не иметь такой аппроксимирующей способности, как блоки LSTM.

Табл. 1. Результаты обучения

алгоритм	параметр	количество пользователей	значение ошибки, м
Дерево решений	time_in_seconds weekday	1	≈ 9,0
	time_in_seconds weekday x_previous z_previous time difference	1	≈ 6,0
Случайный лес	time_in_seconds weekday	1	≈ 10,0
	time_in_seconds weekday x_previous z_previous	1	≈ 2,0
	time_in_seconds weekday	1	≈ 0,4
	time_in_seconds weekday x_previous z_previous time difference	2	≈ 0,7

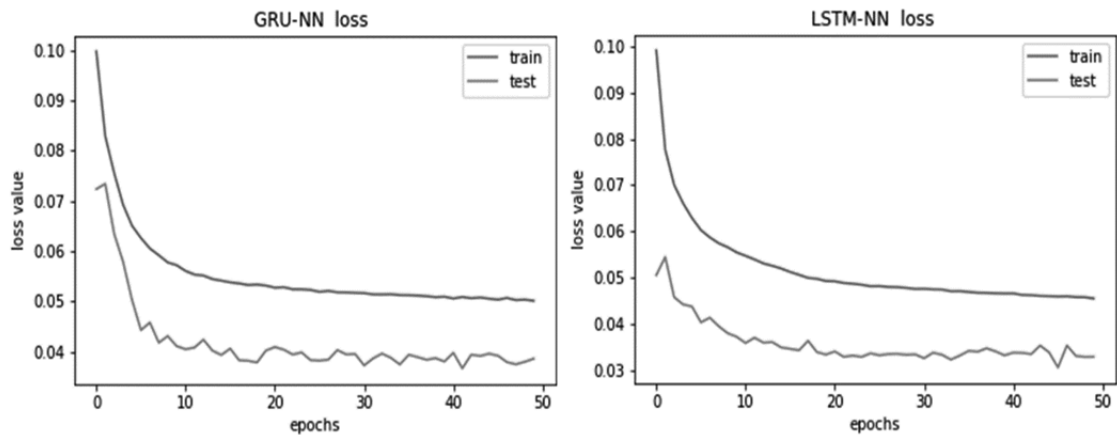


Рис. 4. Результаты тестирования LSTM и GRU-блоков

Две нейронные сети с блоками LSTM и GRU были реализованы и обучены с использованием библиотек Keras и Tensorflow. Разработанные системы состоят из двух скрытых слоев: первый скрытый слой первой нейронной сети является блоком LSTM, второй сети – GRU, второй скрытый слой обеих сетей представлен полносвязным слоем и блоком линейной ректификации (Rectified Linear Unit, ReLU), после чего используется метод исключения (Dropout) для предотвращения переобучения.

Результат обучения реализованных нейронных сетей показан на Рис. 4. Как видно, сеть LSTM имеет более низкие значения СКО для контрольных данных, чем алгоритмы дерева решений и случайного леса.

*D. Глубокое обучение с подкреплением*

Существует несколько современных виртуальных сред, в которых разработчики могут внедрять и обучать модели обучения с подкреплением, а также тестировать свои разработки, например, Openai Gym [22] и Unity Machine Learning Agents (ML-Agents) [23]. В данной рабо-

те было решено использовать Unity ML-Agents по причине ее более широкой функциональности. Unity ML-Agents является плагином Unity с открытым исходным кодом, который позволяет создать среду для обучения интеллектуальных агентов. Реализованы такие механизмы обучения, как обучение с подкреплением, имитационное обучение, нейроэволюция, а также другие методы машинного обучения посредством простого в использовании Python API.

Общая схема процесса обучения агента показана на Рис. 5.

Для обучения и использования агентов плагин Unity ML-Agents должен содержать один подкласс Academy, а также требуемое количество объектов Brain и подклассов Agent. Любые Brain-объекты объект должны быть присоединены к GameObjects, которые являются дочерними классами Academy в иерархии Unity. Экземпляры агента должны быть присоединены к GameObject, который представляет этого агента.

Объекты Brain инкапсулируют процесс принятия решений и должны быть дочерним классом



Рис. 5. Схема глубокого обучения с подкреплением для системы проактивной локализации

Academy в иерархии Unity. Существует два основных типа Brain-объектов: внешние и внутренние: первый используется в процессе обучения, а второй позволяет использовать обученную модель, импортированную в проект Unity.

Класс Agent представляет участника события, который собирает наблюдения и выполняет несколько действий. Данный класс обычно присоединяется к GameObject, который в противном случае является действующим лицом (в нашем случае это пользователь в киберфизическом пространстве). Каждому подклассу Agent должен быть назначен объект Brain. В данной работе интеллектуальный агент был создан путем расширения класса Agent и реализации основных функций:

- Collect Observations() – наблюдение агента за окружающей средой.
- AgentAction() – выполнение действия, выбранного экземпляра класса Brain, и присвоение награды агенту, находящемуся в текущем состоянии.

Agent отправляет собранную информацию в Brain, который использует ее для принятия решения. Когда происходит обучение Agent (или используется уже обученная модель), данные подаются в нейронную сеть как вектор признаков. Чтобы агент мог успешно решить задачу, необходимо предоставить правильную информацию.

Решение, принимаемое модулем Brain, выводится в виде массива действия, переданного функции AgentAction(). Количество элементов этого массива определяется настройками типа и размера объекта Brain: Space Type и Vector Action Space Size. Для создания двух непрерывных управляющих сигналов параметр Space Type был принят непрерывным, Space Size – 2. Первый элемент массива определяет смещение по оси x; второй – смещение по оси z. Кроме того, параметр действия был зафиксирован в диапазоне [-5;5]. Функция Mathf.Clamp() использовалась для остановки алгоритма обучения при вводе больших значений.

Обучение с подкреплением требует вознаграждения, которое должно быть указано в функции AgentAction(). Для определения значений данного параметра использовалась метрика, выраженная в абсолютной ошибке (3). Если значение ошибки составляет меньше 1 метра, система вознаграждает агента 0,4 балла

в противном случае вознаграждение равняется -1 баллу. Результаты обучения, проводимого в течение 60 часов, достаточно противоречивы: наблюдается увеличение значения вознаграждения, при этом увеличивается и величина штрафов. Для выяснения причин этого явления требуются дополнительные исследования.

### 3. Концепция системы проактивной локализации

Как уже упоминалось ранее, важно иметь информацию как о фактическом перемещении пользователя, так и о его потенциальном маршруте в интеллектуальном пространстве. От точности вероятностной оценки его местоположения в пределах предприятия, в котором сформирована КФС, будет зависеть работа инженерных сетей, а также персонифицированных информационных сервисов, предоставляемых в точках доступа к интеллектуальному пространству.

Прогнозирование возможных маршрутов пользователей, основанное на информации об их прошлых перемещениях и фактическом местоположении, поможет более эффективно координировать распределение всех имеющихся ресурсов, так как система будет учитывать фактический спрос на ресурсы (электроэнергию, вентиляцию, ресурсы информационного портала в точках доступа к КФС), а не опираться на заранее установленные усредненные значения. Таким образом, решение проблемы навигации внутри помещений как задачи вероятностного моделирования позволит более эффективно распределять и управлять различными ресурсами в корпоративных системах.

Практическое значение вероятностной системы для предсказания местоположения пользователя КФС заключается в возможности управления ресурсами системы более эффективным способом. Данная система способна прогнозировать маршрут пользователя, отображать свойства, которые могут вызвать наблюдаемое поведение, и использовать информацию о результате события для вывода свойства, которое может повлиять на исход другого события. Использование дополнительных взаимосвязей между элементами КФС позволит в будущем обучать более качественные

математические модели для рекомендательных систем информационного пространства. Концепция системы проактивной локализации показана на Рис. 6.

Пользователь интеллектуального пространства, коим является любая автономная единица, взаимодействующая с системой киберфизического пространства (сотрудник организации, гость, мобильная робототехническая платформа, квадрокоптер и т.д.), в данной системе представляется своей цифровой копией на основе соответствующего профиля, содержащего информацию о биометрических параметрах пользователя, а также данные его персональных идентификаторов (RFID, NFC) и технических характеристиках (в случае с робототехническими единицами). С того момента, когда пользователь распознается в какой-либо локации интеллектуального пространства, его цифровое представление начинает функционировать в цифровом пространстве с использованием ранее реализованной и обу-

ченной нейронной сети. Фактические действия и траектория движения пользователя (светло-серая область на Рис. 5:  $A_{RW}$  – возможные действия пользователя в каждый момент времени (определяется степенью дискретизации наблюдений внутри системы),  $P_{RW}$  – положение пользователя в каждый момент времени, описанное координатами в трехмерном пространстве,  $S_{RW}$  – состояние системы в каждый момент времени), контролируются датчиками системы локализации, одновременно с этим прогнозируется область интеллектуального пространства, где может быть обнаружен пользователь в следующий момент времени (темно-серая область на Рис. 5). После получения информации о фактическом местоположении пользователя и спрогнозированных данных его цифровой копии рассчитывается метрика точности (например, СКО). На основе полученного результата выбирается оптимальная стратегия для алгоритма обучения с целью дальнейшего повышения производительности алгоритма прогнозирования.

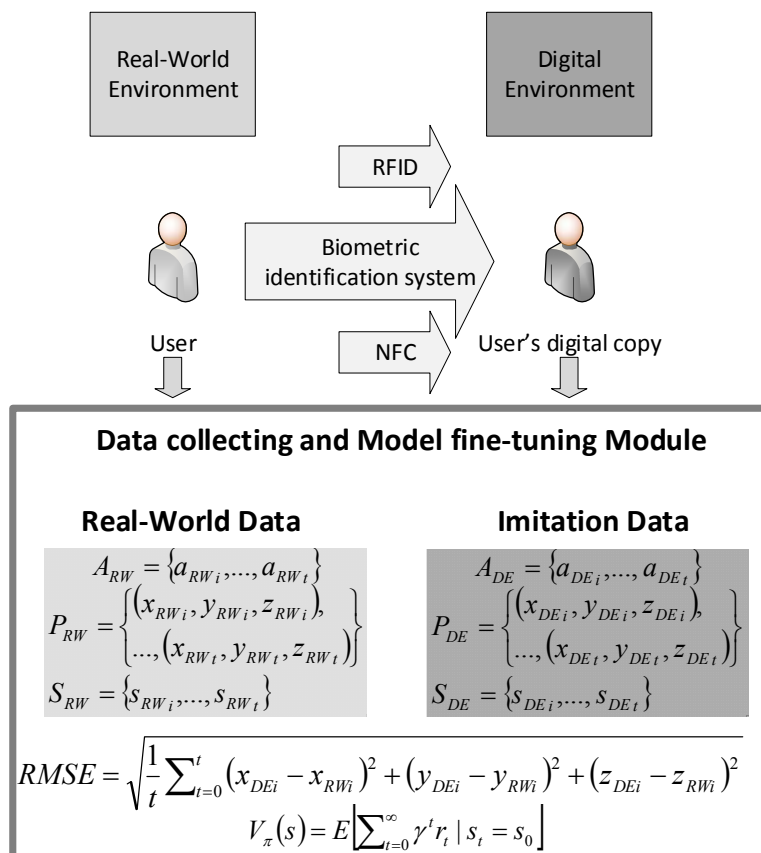


Рис. 6. Концепция системы проактивной локализации



#### 4. Архитектура сервиса проактивной локализации и навигации

При программной реализации киберфизической системы сервиса проактивной локализации и навигации целесообразно использовать микросервисную архитектуру – вариант сервис-ориентированной архитектуры программного обеспечения, предназначенный для разработки насколько это возможно небольших, несвязанных (или со слабой связью) и легко изменяемых независимых модулей – микросервисов. В концепции киберфизической системы каждый сервис, предоставляемый пользователю (система проактивной локализации и навигации, сервис корпоративного телевидения, система видеоконференц-связи и пр.), а также любой функциональный блок, не являющийся полноценным сервисом, но обеспечивающий

при этом корректное функционирование киберфизического пространства в целом (система распознавания лиц, жестов, речи и т.д.), является самостоятельным независимым сервисом. На Рис. 7 представлен возможный вариант архитектуры сервиса проактивной локализации, а также архитектура программной реализации киберфизического пространства.

Для реализации подобной киберфизической системы с множеством сервисов целесообразно использовать оркестровщик с открытым исходным кодом Kubernetes, предназначенный для развертывания и управления контейнеров Linux как единой системой. Kubernetes управляет и запускает docker<sup>1</sup>-контейнеры на большом количестве машин.

Основными сущностями Kubernetes являются:

- **Nodes** – «нода» - это машина в кластере;

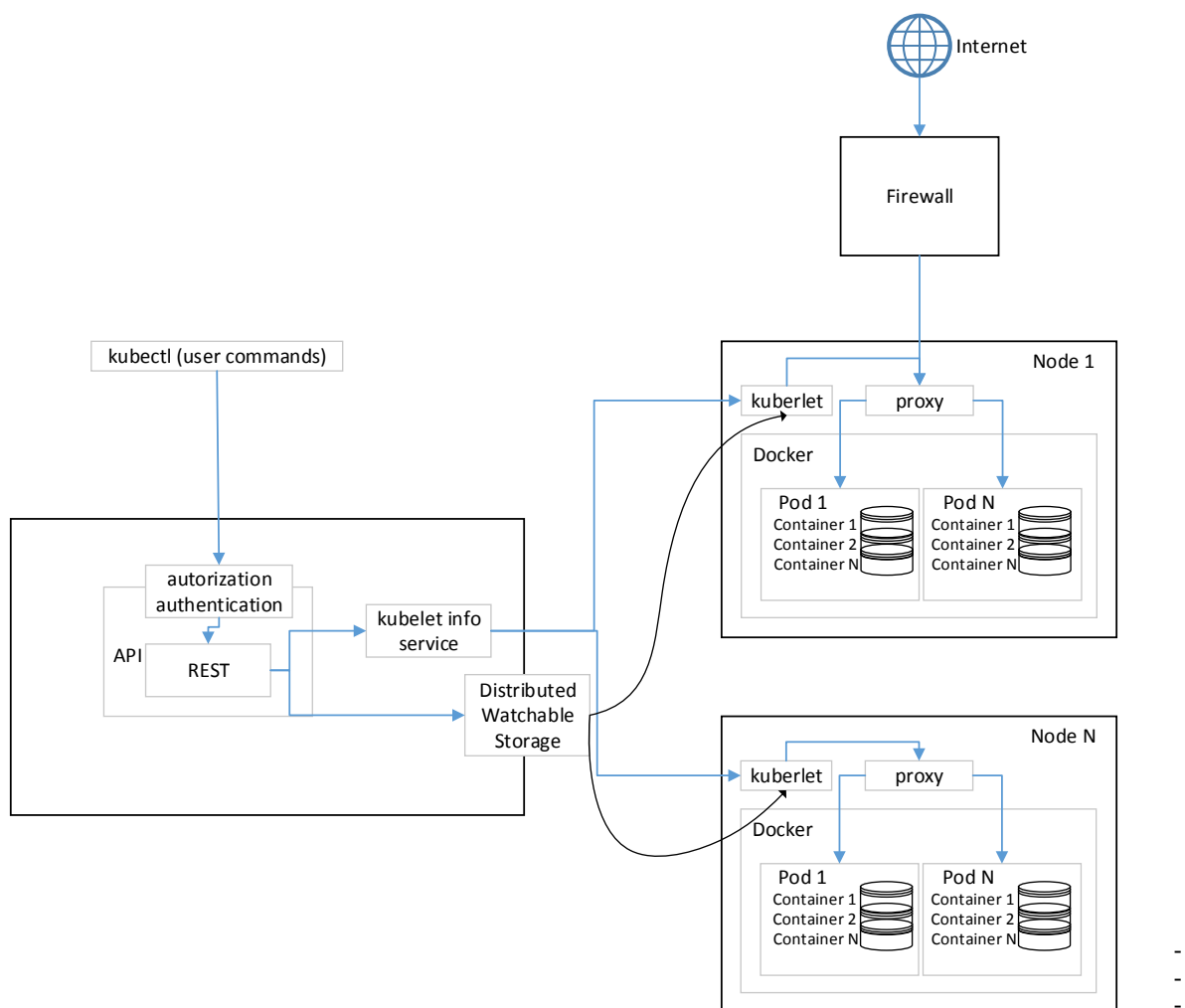


Рис. 7. Архитектура системы киберфизического пространства на основе kubernetes

- **Pods** – одна pod’а – это группа docker-контейнеров с общими разделами, запускаемых как единое целое;
- **Replication Controllers** – контроллер гарантирует, что определенное количество копий pod’ы будут запущены в любой момент времени функционирования кластера;
- **Services** – сервис – это абстракция, определяющая логически объединенный набор pod и политику доступа к ним.
- **Volumes** – раздел – это доступная директория в контейнере.
- **Labels** – лейблы – это пары «ключ-значение», которые закрепляются за объектами, например, pod’ами. Лейблы могут быть использованы для создания и выбора набора объектов.
- **Kubectl Command Line Interface** – kubectl интерфейс командной строки для управления Kubernetes.

Функционирующий кластер состоит из агента (kubelet), запущенного на машинах, а также компонентов мастера (APIs, scheduler и т.д.), поверх решения с распределенным хранилищем. На каждой рабочей машине (pod’е) запускаются сервисы, которые необходимы для управления этой машиной со стороны мастера и для запуска приложений. При этом на каждой машине запускаются docker-контейнеры, внутри которых функционирует конкретный сервис, либо обособленный функциональный блок, общий для нескольких сервисов. Агент (kubelet) управляет pod’ами, их контейнерами, образами, разделами и т.д. Кроме того, на каждой рабочей станции запускается проху-балансировщик (kube-проху), выполняющий перенаправление потоков TCP и UDP между элементами backend’а.

Система управления kubernetes обеспечивает работу api-сервера, являющегося в то же время CRUD сервером со встроенной бизнес-логикой, реализованной посредством отдельных внешних компонентов или через внутренние плагины.

Все остальные функции на уровне кластера представлены в Controller Manager. Например, ноды обнаруживаются, управляются и контролируются средствами node controller. Эта сущность может быть разделена на отдельные компоненты, чтобы сделать их независимо подключаемыми.

## Заключение

В данной работе представлены результаты анализа методов и алгоритмов прогнозирования последовательностей и временных рядов: Decision Trees, Random Forest, искусственные нейронные сети с блоками LSTM и GRU. Описана концепция системы проактивной локализации, являющейся частью КФС, определена стратегия взаимодействия пользователя с интеллектуальным пространством и его цифровое представление в среде моделирования. Дальнейшие исследования будут направлены на разработку полнофункциональной платформы КФС с набором сервисов, включая систему проактивной локализации.

## Литература

1. Смирнов А.В., Левашова Т.В., Принципы и модели контекстно-управляемой интеграции знаний, Информационные технологии и вычислительные системы, № 4, 2013, с. 58-73.
2. Кашевник А. М., Подход к обеспечению семантической интероперабельности мобильных роботов при формировании коалиций, Информационные технологии и вычислительные системы, № 1, 2017, с. 90-100.
3. Levonevskiy D., Vatamaniuk I., Saveliev A., Integration of Corporate Electronic Services into a Smart Space Using Temporal Logic of Actions, International Conference on Interactive Collaborative Robotics, Springer, Cham, 2017, pp. 134-143.
4. Lee J., Bagheri B., Kao H.A. A cyber-physical systems architecture for industry 4.0-based manufacturing systems, Manufacturing Letters, 2015, vol. 3, pp. 18-23.
5. Amri M.-H., Becis Y., Aubry D., Ramdani N., Indoor human/robot localization using robust multi-modal data fusion, IEEE International Conference on Robotics and Automation (ICRA), 2015.
6. Liu Z., Yang D.-S., Wen D., Zhang W.-M., Mao W., Cyber-Physical-Social Systems for Command and Control, IEEE Intelligent Systems, July-Aug. 2011, vol. 26, i. 4, pp. 92-96.
7. Frazzon E.M., Hartmann J., Makuschewitz T., Scholz-Reiter B., Towards Socio-Cyber-Physical Systems in Production Networks, Procedia CIRP, 2013, vol. 7, pp. 49-54.
8. Shi J., Wan J., Yan H., Suo H. A survey of Cyber-Physical Systems, International Conference on Wireless Communications and Signal Processing (WCSP), 2011.
9. Balico L.N., Loureiro A.A.F., Nakamura E.F., Barreto R.S., Pazzi R.W., Oliveira H.A.B.F., Localization Prediction in Vehicular Ad Hoc Networks, IEEE Communications Surveys & Tutorials (Early Access), 2018.
10. Nadembega A., Hafid A., Taleb T., A Destination and Mobility Path Prediction Scheme for Mobile Networks, IEEE Transactions on Vehicular Technology vol. 64, i. 6, June 2015, pp. 2577-2590.

11. Lin K., Chen M., Deng J., Hassan M. M., Fortino G., Enhanced Fingerprinting and Trajectory Prediction for IoT Localization in Smart Buildings, *IEEE Transactions on Automation Science and Engineering*, July 2016, vol. 13, i. 3, pp. 1294-1307.
12. Shit R.C., Sharma S., Puthal D., Zomaya A.Y., Location of Things (LoT): A Review and Taxonomy of Sensors Localization in IoT Infrastructure, *IEEE Communications Surveys & Tutorials (Early Access)*, 2018.
13. Alletto S., Cucchiara R., Fiore G.D., Mainetti L., Mighali V., Patrono L., Serra G., An Indoor Location-Aware System for an IoT-Based Smart Museum, *IEEE Internet of Things Journal*, April 2016, vol. 3, i. 2, pp. 244-253.
14. Pahlavan K., Krishnamurthy P., Geng Y., Localization Challenges for the Emergence of the Smart World, *IEEE Access*, 2015, vol. 3, pp. 3058-3067.
15. Drevelle V., Bonnifant P., Robust positioning using relaxed constraint-propagation, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010.
16. Cook D. J., Schmitter-Edgecombe M., Dawadi P., Analyzing Activity Behavior and Movement in a Naturalistic Environment Using Smart Home Techniques, *IEEE Journal of Biomedical and Health Informatics*, 2015, vol.19, i. 6, pp. 1882-1892.
17. Dixit A., Naik A., Use of prediction algorithms in smart homes, *International Journal of Machine Learning and Computing*, 2014, 4 (2), p. 157.
18. Wang Y., Yuan N.J., Lian D., Xie X., Chen E., Rui Y. Regularity and Conformity: Location Prediction Using Heterogeneous Mobility Data, *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, pp. 1275-1284.
19. Xu G., Gao S., Daneshmand M., Wang C., Liu Y. A Survey for Mobility Big Data Analytics for Geolocation Prediction, *IEEE Wireless Communications*, 2017, vol. 24, i. 1, pp. 111-119.
20. Kim Y., An J., Lee M., Lee Y. An Activity-Embedding Approach for Next-Activity Prediction in a Multi-User Smart Space, *IEEE International Conference on Smart Computing (SMARTCOMP)*, 2017.
21. Qolomany B., Al-Fuqaha A., Benhaddou D., Gupta A. Role of Deep LSTM Neural Networks And Wi-Fi Networks in Support of Occupancy Prediction in Smart Buildings, *The 15th IEEE International Conference on Smart City*, 2017, pp. 18-20.
22. Saveliev A., Malov D., Edemskii A. Proactive Localization System Concept for Users of Cyber Physical Space, *International Conference on Interactive Collaborative Robotics. – Springer, Cham*, 2018. (In press).
23. Official documentation of GYM - a toolkit for developing and comparing reinforcement learning algorithms. Available at: <https://gym.openai.com/> (accessed September 28, 2018).
24. Official documentation of ML-Agents in Unity3D. Available at: <https://unity3d.com/ru/machine-learning> (accessed September 28, 2018).

**Малов Дмитрий Александрович.** Федеральное бюджетное учреждение науки Санкт-Петербургского института информатики и автоматизации РАН (СПИИРАН). Младший научный сотрудник. Аспирант. Область научных интересов: машинное обучение, нейронные сети, теория оптимизации, адаптивное управление, интеллектуальные пространства. E-mail: malovdmitrij@gmail.com

**Едемский Александр Юрьевич.** Федеральное бюджетное учреждение науки Санкт-Петербургского института информатики и автоматизации РАН (СПИИРАН). Программист. Магистр Санкт-Петербургского государственного электротехнического университета «ЛЭТИ» имени В. И. Ульянова (Ленина). Область научных интересов: машинное обучение, нейронные сети, киберфизические системы. E-mail: qapple@ua.gu

**Савельев Антон Игоревич.** Федеральное бюджетное учреждение науки Санкт-Петербургского института информатики и автоматизации РАН (СПИИРАН). Старший научный сотрудник. Старший преподаватель кафедры электромеханики и робототехники института инновационных технологий в электромеханике и робототехнике Университета ГУАП. Количество печатных работ: более 30. Область научных интересов: робототехника, обработка цифровых сигналов, машинное обучение. E-mail: saveliev@iias.spb.su

## Development of a system of proactive localization of the cyber-physical space based on machine learning methods

D.A. Malov, A.Y. Edemsky, A. I.Saveliev

St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences, Saint-Petersburg, Russia

Cyberphysical systems (CFS) are based on the seamless integration of computing power into the physical environment within an organization, enterprise, or production. In this paper we propose a system of proactive localization for tracking and forecasting the location of users and mobile robots. The developed system allows to predict the activity of the monitored object using various methods of machine learning. The paper presents comparative analysis of various machine learning models, as well as the concept of a proactive localization system.

**Keywords:** Cyberphysical system, time series forecasting, proactive localization, recurrent neural networks, reinforcement learning.

DOI 10.14357/20718632180408

## References

1. Smirnov A.V., Levashova T.V., Foundations and models of context-aware knowledge integration, Information technology and computer systems, vol. 4, 2013, pp. 58-73. (in Russian)
2. Kashevnik A.M., An approach to semantic interoperability support between mobile robots for coalition formation, Information technology and computer systems, vol. 1, 2017, pp. 90-100. (in Russian)
3. Levonevskiy D., Vatamaniuk I., Saveliev A., Integration of Corporate Electronic Services into a Smart Space Using Temporal Logic of Actions, International Conference on Interactive Collaborative Robotics, Springer, Cham, 2017, pp. 134-143.
4. Lee J., Bagheri B., Kao, H.A. A cyber-physical systems architecture for industry 4.0-based manufacturing systems, Manufacturing Letters, 2015, vol. 3, pp. 18-23.
5. Amri M.-H., Becis Y., Aubry D., Ramdani N., Indoor human/robot localization using robust multi-modal data fusion, IEEE International Conference on Robotics and Automation (ICRA), 2015.
6. Liu Z., Yang D.-s., Wen D., Zhang W.-m., Mao W., Cyber-Physical-Social Systems for Command and Control, IEEE Intelligent Systems, July-Aug. 2011, vol. 26, i. 4, pp. 92-96.
7. Frazzon E. M., Hartmann J., Makuschewitz T., Scholz-Reiter B., Towards Socio-Cyber-Physical Systems in Production Networks, Procedia CIRP, 2013, vol. 7, pp. 49-54.
8. Shi J., Wan J., Yan H., Suo H., A survey of Cyber-Physical Systems, International Conference on Wireless Communications and Signal Processing (WCSP), 2011.
9. Balico L. N., Loureiro A. A. F., Nakamura E. F., Barreto R.S., Pazzi R. W., Oliveira H. A. B. F., Localization Prediction in Vehicular Ad Hoc Networks, IEEE Communications Surveys & Tutorials (Early Access), 2018.
10. Nadembega A., Hafid A., Taleb T., A Destination and Mobility Path Prediction Scheme for Mobile Networks, IEEE Transactions on Vehicular Technology vol. 64, i. 6, June 2015, pp. 2577-2590.
11. Lin K., Chen M., Deng J., Hassan M. M., Fortino G., Enhanced Fingerprinting and Trajectory Prediction for IoT Localization in Smart Buildings, IEEE Transactions on Automation Science and Engineering, July 2016, vol. 13, i. 3, pp. 1294-1307.
12. Shit R.C., Sharma S., Puthal D., Zomaya A. Y., Location of Things (LoT): A Review and Taxonomy of Sensors Localization in IoT Infrastructure, IEEE Communications Surveys & Tutorials (Early Access), 2018.
13. Alletto S., Cucchiara R., Fiore G. D., Mainetti L., Mighali V., Patrono L., Serra G., An Indoor Location-Aware System for an IoT-Based Smart Museum, IEEE Internet of Things Journal, April 2016, vol. 3, i. 2, pp. 244-253.
14. Pahlavan K., Krishnamurthy P., Geng Y., Localization Challenges for the Emergence of the Smart World, IEEE Access, 2015, vol. 3, pp. 3058-3067.
15. Drevelle V., Bonnifait P., Robust positioning using relaxed constraint-propagation, IEEE/RSJ International Conference on Intelligent Robots and Systems, 2010.
16. Cook D.J., Schmitter-Edgecombe M., Dawadi P., Analyzing Activity Behavior and Movement in a Naturalistic Environment Using Smart Home Techniques, IEEE Journal of Biomedical and Health Informatics, 2015, vol.19, i. 6, pp. 1882-1892.
17. Dixit A., Naik A., Use of prediction algorithms in smart homes, International Journal of Machine Learning and Computing, 2014, 4 (2), p. 157.
18. Wang Y., Yuan N.J., Lian D., Xie X., Chen E., Rui Y., Regularity and Conformity: Location Prediction Using Heterogeneous Mobility Data, Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2015, pp. 1275-1284.
19. Xu G., Gao S., Daneshmand M., Wang C., Liu Y., A Survey for Mobility Big Data Analytics for Geolocation Prediction, IEEE Wireless Communications, 2017, vol. 24, i. 1, pp. 111-119.
20. Kim Y., An J., Lee M., Lee Y., An Activity-Embedding Approach for Next-Activity Prediction in a Multi-User Smart Space, IEEE International Conference on Smart Computing (SMARTCOMP), 2017.
21. Qolomany B., Al-Fuqaha A., Benhaddou D., Gupta A., Role of Deep LSTM Neural Networks And Wi-Fi Networks in Support of Occupancy Prediction in Smart Buildings, The 15th IEEE International Conference on Smart City, 2017, pp. 18-20.
22. Saveliev A., Malov D., Edemskii A. Proactive Localization System Concept for Users of Cyber Physical Space, International Conference on Interactive Collaborative Robotics. – Springer, Cham, 2018. (unpublished)
23. Official documentation of GYM - a toolkit for developing and comparing reinforcement learning algorithms. Available at: <https://gym.openai.com/> (accessed September 28, 2018).
24. Official documentation of ML-Agents in Unity3D. Available at: [https://unity3d.com/ru/machine-learning\\_](https://unity3d.com/ru/machine-learning_) (accessed September 28, 2018).

**D.A Malov.** Junior researcher of St.Petersburg Institute for Informatics and Automation of the Russian academy of science (SPIIRAS), St.Petersburg, Russia, malovdmitrij@gmail.com

**A.Y. Edemsky.** Programmer of St.Petersburg Institute for Informatics and Automation of the Russian academy of science (SPIIRAS), St.Petersburg, Russia, qapple@ya.ru

**A.I. Saveliev.** Senior researcher of St.Petersburg Institute for Informatics and Automation of the Russian academy of science (SPIIRAS), St.Petersburg, Russia, saveliev@iias.spb.su