Bachelor's thesis

Machine Learning-Based User Movement Prediction in Layer 2 Networks

Vorhersage von Benutzerbewegungen in Layer 2 Netzen basierend auf Maschinellem Lernen

by
Lina Wilske

Supervisors
Prof. Dr. Holger Karl
Leonard Paeleke
Internet Technology and Softwarization Group

Hasso Plattner Institute at University of Potsdam

August 24, 2023

Abstract

- human movement not considered when roaming of mobile devices appears
- could lead to many unnecessary handovers and thus to bad performance
- this thesis proposes a machine learning-based approach to predict the Access Point (AP) a user is nearest to based on the Received Signal Strength Indication (RSSI) of the APs

  IS nearest to? rahter: *will be* nearest to?

- the chosen machine-learning model is Long Short-Term Memory (LSTM) and is trained on real-world data from a dataset of a competition by Microsoft Research

  in thext, make sure to explain WHY lstm was chosen.

- the dataset analysis showed that interpolation is necessary to ensure that Wireless Fidelity (Wi-Fi) and waypoint data together are used in the prediction
- many APs are deployed in the buildings of the dataset, which makes the prediction task hard
- one building and one floor were chosen to be the dataset for this thesis
- the evaluation of the model shows that the mode predicts the AP a user is nearest to with a top 3 accuracy of about 71%
- too many classes, which makes prediction too hard and top 3 accuracies too low
- with fewer classes, the model could predict better
- in the future, data explicitly generated for this purpose could be used to predict the top k access points so that the location of the access points is known and can be considered in prediction

Contents

1 Introduction

In large-scale Wi-Fi environments such as office buildings, shopping malls, and airports, where multiple APs are required, people often move around indoors with their mobile devices. To maintain a stable connection to the Service Set Identifier (SSID), the station must remain in the range of the AP or may roam to another AP with the same SSID. However, the current roaming process in 802.11k/r [7][6] does not include human movement.

not sure how a roaming process could INCLUDE human movement? consider, take into account, optimize using that information , ... ?

For example, if a user's station is moving away from AP1 towards AP2 and further towards AP3, ideally, AP3 should not initiate the roaming process but instead AP2 and then AP3 is connected.

not sure why this is IDEAL?

An AP may instruct a client device to roam based on signal strength without considering the device's trajectory or the user's likely destination. Real-time applications such as video conferencing are particularly sensitive to these hand-offs, which may result in dropped connections and unsatisfied users.

This thesis will explore if a time-series Machine Learning (ML) model can predict the nearest AP to which a station may connect next. This nearest station needs to be in the top 3 of the predictions to be considered a correct prediction.

why? ideally, I want THE next station. Top 3 is just a relexation of that requirement; needs to be explained. That needs a more extensive discussion. Either here (I think), or alternatively when you get to the design section, at the latest at the evaluation section

Hence, this thesis needs data with Wi-Fi, waypoint of clients, and sensor data, e.g., acceleration.

why "hence"? Why does that follow from the top-3 requirement?

new thought, new paragraph

A time-series ML model requires as input time-series data.

maybe briefly explain what that is?

There are two possible data sources: generate new or utilize existing data. Data generation needs a comprehensive plan for accounting data setup and collection.

and to makes ure that this is representative, ...

This process is time-consuming and needs a lot of planning and evaluation beforehand, which is not the focus of this thesis. Thus, we

royal plural??

will utilize pre-existing data from a 2021 competition by Microsoft Research[8].

it does not need long explanation why you used existing data sources. but why THIS one, what were the criteria you used to select, what were the alternative traces? Need not be discussed in intro, can be alter, but needs a reference that section, if discussed later.

The data will be analyzed in Chapter 3

cref should capitalizue Chapter 3 . Fixed in preamble

to determine what parts of the data we will use for the ML model. Additionally, the data will be prepared for a time series ML model.

reference to chapter 2 is missing? gap in intro narrative

After that, we will discuss the suitability of some pre-selected time series ML models for the task in Chapter 4.

aha! a bit late in intro story line... but ok

Due to many data,

?? not sure what that means

we will discover in Chapter 3,

no “the” – Chapter 4 is a proper name. You would also not write “we will discover the Peter”, but “disciver Peter”

this thesis will implement, in Chapter 5, a ML model, train and test it for one site and floor of the competition. Finally, in Chapter 7, we will evaluate the model’s performance and conclude if this prediction could be useful.

2 Background

2.1 Machine Learning

if there is a 2.1, there must be also a 2.2

2.1.1 Time Series Prediction

Time series data consists of data points arranged chronologically, prevalent in numerous domains like stock prices. Due to its inherent temporal dependencies, where subsequent data points influence previous ones, specific machine learning techniques are applied. These include Multilayer Perceptron (MLP), Hidden Markov Model (HMM), and Recurrent Neural Network (RNN) models such as LSTM. Each model is designed to capture and leverage temporal patterns within the data, predicting future trends based on historical observations. [14]

2.1.2 Hyperparameter tuning

In machine learning, hyperparameters play a vital role in model development. These are parameters such as the learning rate, neural network layers, and the number of windows or batch sizes. Proper selection of hyperparameters, known as hyperparameter tuning or optimization, is crucial to optimize model performance. This iterative procedure involves exploring various hyperparameter combinations for the configuration that yields the most accurate predictions. Hyperparameters can be tuned by, e.g., random search, which can be done manually or using libraries such as keras-tuner[15].

2.1.3 Classification

Classification models in ML predict specific categories or classes for input data. By training on input features and labels, these models categorize unseen data. They find use in many domains, producing outputs such as spam or not spam, positive or negative sentiment, and malignant or benign tumors. A specific type of classification is multi-class classification which categorizes more than two classes. [1, pp.179-182]

2.1.4 Univariate and Multivariate Time Series

- univariate: one observation recorded sequentially over time, e.g., temperature, stock prices; focus on understanding and forecasting a single variable's behavior
- multivariate: multiple observations recorded over the same time intervals, allowing for the analysis of interrelationships and interdependencies between these variables,

e.g., temperature and humidity; focus on delving into understanding dynamic interactions and co-movements between multiple variables

2.1.5 Temporal Dependency Handling

Temporal dependency handling refers to the ability of a ML model to recognize and leverage the relationships or dependencies between data points that are separated by time. In time series data, previous values can influence the value at a given time point, and understanding this dependency is crucial for accurate predictions.

this is REALLY brief. and none of that really matters all that much; you are not supposed to write a textbook here! What is missing is the RELATED work: what other handover prediction papers are there; how well did they work; similar approach or different approach; what is the expectation levels of results; etc. You need something against which you are comparing your own results. This is really IMPORTANT

3 Dataset analysis and preparation

The dataset used in this thesis is the Indoor Location & Navigation from kaggle[12], which was part of a competition of Microsoft Research in 2021[8].

again: WHY this one? you always need to give reasons for such decisions!

The company XYZ[10]

?? that is really their name?

recorded the data in shopping malls and was provided by Microsoft Research for this competition. The goal for the competition was, given a site-path file,

what is that?

predict the floor and waypoint locations at a timestamp given in the submission files. In the following, the dataset and data will be analyzed.

3.1 Components of the dataset

As noted in the kaggle notebook "Indoor Navigation: Complete Data Understanding" [10] the data consists of 3 parts:

- a train folder with train path files, organized by site and floor
- a test folder with test path files, organized by site and floor but without waypoint data
- a metadata folder with floor metadata, organized by site and floor, which includes floor images, further information, and a geojson map

The train folder contains 204 subfolders, which represent each site where the data was recorded. In each site folder are a minimum of one and a maximum of twelve subfolders, which represent the floors of the site; the median is five floors. Overall there are 26,925 files, each containing the movement of one person for a specific site and floor. Per floor, there are between one and 284 files with a median of 14. The floor F1 of the site 银泰城(城西店), which was hashed as "5d27075f03f801723c2e360f" in the train folder of the competition, has the most files.

For this thesis, the submission files, as well as the test folder, will not be used because our goal is not to predict the floor and site name for a certain timestamp but to predict the Basic Service Set Identifier (BSSID) to which a device may connect next. Therefore, we will not analyze the content of these folders in more detail.

no submission files are used? I am confused. What DO you use, then?

3.2 File structure

Each file in each floor folder is a **.txt** file. The first two lines and the last are denoted with "#". The first line

?

contains the start time of the recording, the second site information SiteID as hash, SiteName, FloorId as hash, and FloorName. The last line contains the end time of the recording. The main part of the data consists of the collected data. Each line contains a UNIX timestamp in milliseconds, followed by a data type and the data itself, which are all separated by a tabulator. The GitHub repository of the competition[9] shows that the data type in the second column followed by its data can be one of the following:

(1) TYPE_ACCELEROMETER with x, y and z acceleration and an accuracy value

(2) TYPE_MAGNETIC_FIELD with x, y and z magnetic field and an accuracy value

(3) TYPE_GYROSCOPE with x, y and z gyroscope and an accuracy value

(4) TYPE_ROTATION_VECTOR with x, y and z rotation vector and an accuracy value

(5) TYPE_MAGNETIC_FIELD_UNCALIBRATED with x, y and z magnetic field and an accuracy value

(6) TYPE_GYROSCOPE_UNCALIBRATED with x, y and z gyroscope and an accuracy value

(7) TYPE_ACCELEROMETER_UNCALIBRATED with x, y and z acceleration and an accuracy value

(8) TYPE_WIFI with SSID, BSSID, RSSI, frequency, and last seen timestamp of the access point. The SSID and BSSID are hashed.

(9) TYPE_BEACON with Universally Unique Identifier (UUID), Major Identifier (MajorID), Minor Identifier (MinorID), Transmission Power (TxPower), RSSI, distance to the device measured by the beacon, Media Access Control (MAC) address and a timestamp as padding data. The MajorID and MinorID are hashed.

(10) TYPE_WAYPOINT with x and y coordinates which are the ground truth location labeled by the surveyor

captions for tables, listings etc. usually go above the table, ... Usually, only figure cpations go below

```
#   startTime:1571462193934
#   SiteID:5d27099303f801723c32364d SiteName:银泰百货(庆春
     店) FloorId:5d27099303f801723c323650 FloorName:4F
1571462193944   TYPE_WAYPOINT   57.885998   69.501526
1571462194071   TYPE_ACCELEROMETER  -0.95254517 0.7944031   8.928757    2
1571462194071   TYPE_MAGNETIC_FIELD -25.65918   -4.4784546  -28.201294  3
1571462194071   TYPE_GYROSCOPE  -0.22373962 -0.07733154 -0.16847229 3
1571462194071   TYPE_ROTATION_VECTOR    0.04186145  -0.02101801 -0.72491926 3
1571462194071   TYPE_MAGNETIC_FIELD_UNCALIBRATED    -4.8568726  10.406494   -387.44965  20.802307
     14.884949   -359.24835  3
1571462194071   TYPE_GYROSCOPE_UNCALIBRATED -0.22218323 -0.068359375    -0.1628418  0.0026245117
     9.765625E-4 -7.6293945E-4   3
1571462194071   TYPE_ACCELEROMETER_UNCALIBRATED -0.95254517 0.7944031   8.928757    0.0 0.0 0.0 3
...
```

```
1571462194883   TYPE_WIFI   b06c4e327882fab58dfa93ea85ca373a54e887b5    9
     f967858afcbb907af6e5adef766c7e7b936ef07    -63 2462    1571462190744
1571462194883   TYPE_WIFI   8204870beb9d02995dab3f08aad97af5eab723cc    0413
     b35df78fc865af15b4721d5aeb33ff57da45    -64 2447    1571462188686
...
1571462194020   TYPE_BEACON 07efd69e3167537492f0ead89fb2779633b04949
     b6589fc6ab0dc82cf12099d1c2d40ab994e8410c    76e907e391ad1856762f70538b0fd13111ba68cd    -57 -71
      5.002991815535578   1b7e1594febd760b00f1a7984e470867616cee4e    1571462194020
...
1571462195943   TYPE_WAYPOINT   59.72475    69.02152
#   endTime:1571462195976
```

Listing 3.1: A snippet from the dataset of the file 5daa9e38df065a00069beb79.txt of the floor F4 of the site with the ID 5d27099303f801723c32364d

das pinyin für den chinesischen Namen ist: Yíntài chéng (chéngxi dìan) Yintai city (chengxi branch)

Each file contains a different amount of waypoints and sensor data. The first and last data type in each file is a Type (10). Lines with types from (1) to (7) occur every 20 ms and are measured at the same time. (8) occurs about every 1800-2200 ms. (10) data is not evenly distributed. An assumption for this is that the recording of the waypoint data is triggered by an exterior event, e.g., a button press. As seen in Listing 3.1, the data are measured separately from each other, so there are no combinations of the data types.

A prediction of the next BSSID will only work per site due to the different architectures of the sites.

I think I can guess, but not really sure what you mean?

But even when limiting a prediction model to a single site, prediction could be

?? “could be” ist immer richtig

difficult for a whole site because the APs are different on each floor, which may result in many APs for the prediction.

versteh ich nihct...

To get better results in the prediction,

Viel schöner: To better predict, ...

we will focus on a single floor of a site. Table 3.1

Eigenname; kein the

shows an analysis of the site with the most files for a single floor.

not toally clear what you are trying to tell me

3.3 Improving data for an ML model

Ist “improving” hier richtig? “preprocessing”?

As seen in previous sections, a location for the time of *TYPE_WIFI*

warum math mode??? Korrekter wäre texttt?

data points is not provided. Also, we only

Information	Value
Total data points	7,157,081
Average data points per file	25,201
Number of waypoints	2,027
Lines of each (1) to (7) data	746,689
Lines of Wi-Fi data	1,862,044
Lines of beacon data	66,187
Number of BSSIDs	4,795
Number of APs	4,795
Number of SSIDs	1,421
RSSI range	-93 to -13 dBm

Table 3.1: Summary of data for F1 of site 银泰城(城西店)

?? da sist ne Menge

have 2,027 waypoints for this floor but 1,862,044 lines of Wi-Fi data, as seen in Table 3.1. The visualization of the waypoints can be seen in Fig. 3.1.

Further human movement may have occurred between *TYPE_WAYPOINT* and *TYPE_WIFI* data. However, they can be combined using linear interpolation, as seen in Fig. 3.2.

Thus, *TYPE_WAYPOINT* and *TYPE_WIFI* are combined to get a location for the Wi-Fi data point.

?? versteh ich nicht. Wie? Details? WArum ist das angemessen; kam das raus, was Sie erwartet hatten?

The interpolation results in 6549 waypoints, which is three times more than the original amount of waypoints, as seen in Fig. 3.3. Furthermore, we will also interpolate *TYPE_ACCELEROMETER* to improve the predictions. Now a multivariate time series is interpolated out of the original data, which will be used for the machine learning model. A further interpolation is possible, but we will focus on a simpler multivariate time series in this thesis.

simpler? compared to what? Sorry, aber das ist hier schwer zu folgen. Da müssen mehr Details hin.

3.4 Peculiarities of the data

Further analysis of the dataset revealed some peculiarities, which are described in the following.

The data is collected by different devices at different timestamps and days. A problem for the time series is that the waypoint data were measured irregularly.

- As in Fig. 3.1, some waypoints seem to be very distant from the next one
- Listing 3.2 shows the top 10 pairs of waypoints with the most significant metric differences

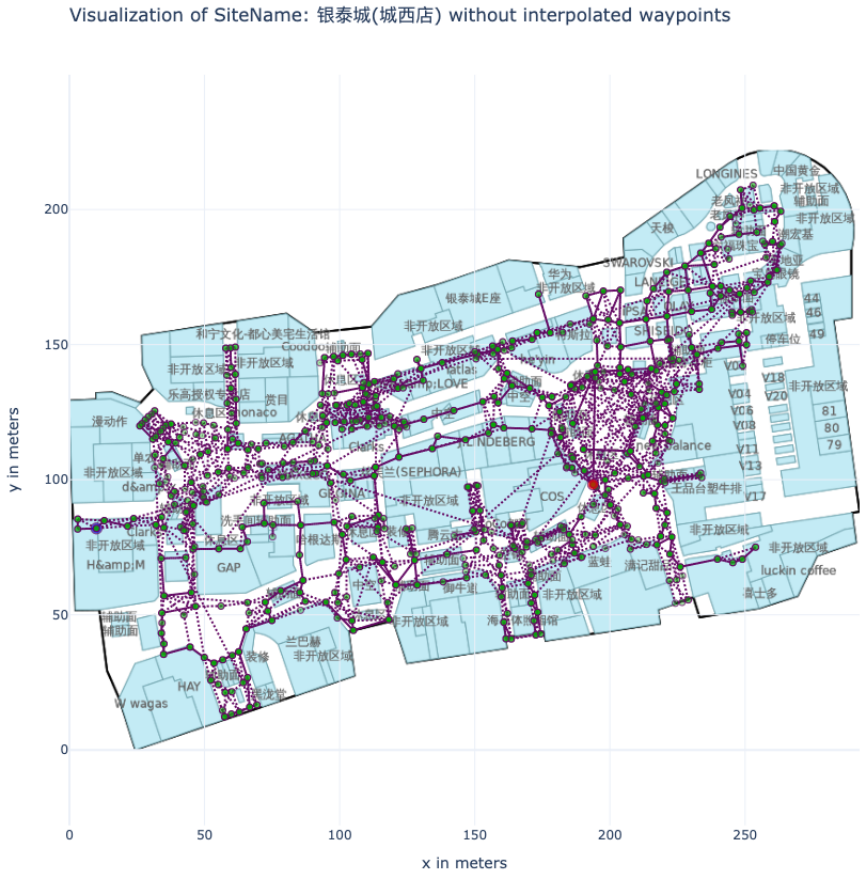

Figure 3.1: Visualization of the 2,027 waypoints of the site 银泰城(城西店)

- if points are more than 10 meters apart, we will define them as "to apart from each other."
- split up the data where the distance between two points is more than 10 meters
- result: 123 files, with interpolation

ok, but what follows from that, WHY is that important; WHY is it a problem or opportunity, ...? Don't just describe the mere facts; interpret them, explain them!

1. Point 1: (247.96523998265695, 168.7631635050295), Point 2: (117.92375106521739, 51.997759545341616), Metric Difference: 174.77113148839442
2. Point 1: (98.66346, 127.5971), Point 2: (258.75049789436116, 181.23350740899357), Metric Difference: 168.83342057049657
3. Point 1: (189.58672, 71.454666), Point 2: (89.73448203762376, 102.255128190099), Metric Difference: 104.49467879858156
4. Point 1: (223.49295, 145.0939), Point 2: (174.26284532732006, 78.86335505811792), Metric Difference: 82.52325908119289
5. Point 1: (34.864815, 35.45561), Point 2: (33.284438514193546, 110.76117936967742), Metric Difference: 75.32215057954856
6. Point 1: (50.31085719185683, 92.03105531572366), Point 2: (114.97229034709193, 123.04521228267667), Metric Difference: 71.71456525741291

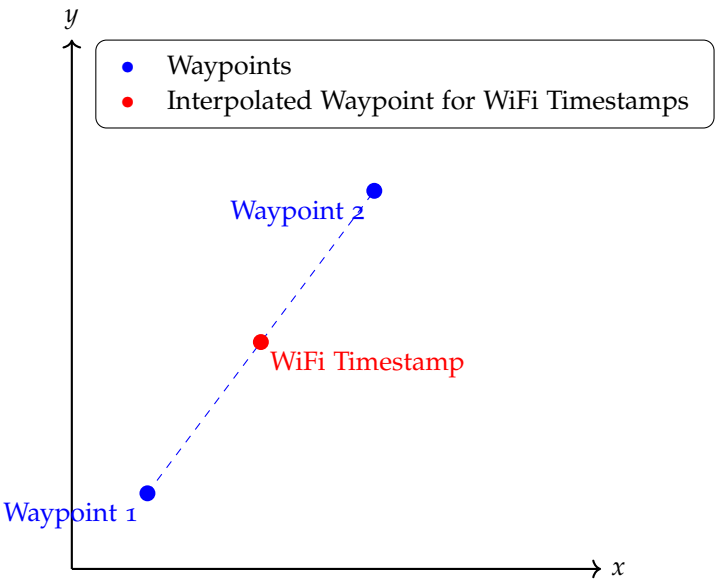

Figure 3.2: Visualization of linear interpolation for Wi-Fi timestamps based on given waypoints. The blue points represent the original waypoints, while the red points show the interpolated positions for specific Wi-Fi

```
7. Point 1: (150.91972285390713, 145.15169976783693), Point 2: (222.23440919809525,
     146.11043967333333), Metric Difference: 71.32113060360388
8. Point 1: (64.64140228107132, 25.345204272946134), Point 2: (34.47475562023909, 84.44615420072283)
     , Metric Difference: 66.35471990088624
9. Point 1: (56.83799274253731, 74.52090035349569), Point 2: (94.43750948519768, 125.97292215289488)
     , Metric Difference: 63.72624425248555
10. Point 1: (172.3439286324042, 56.200716101045295), Point 2: (212.4478039192399,
     99.33967479470648), Metric Difference: 58.90068395354572
```

Listing 3.2: Top 10 pairs with the most significant metric differences

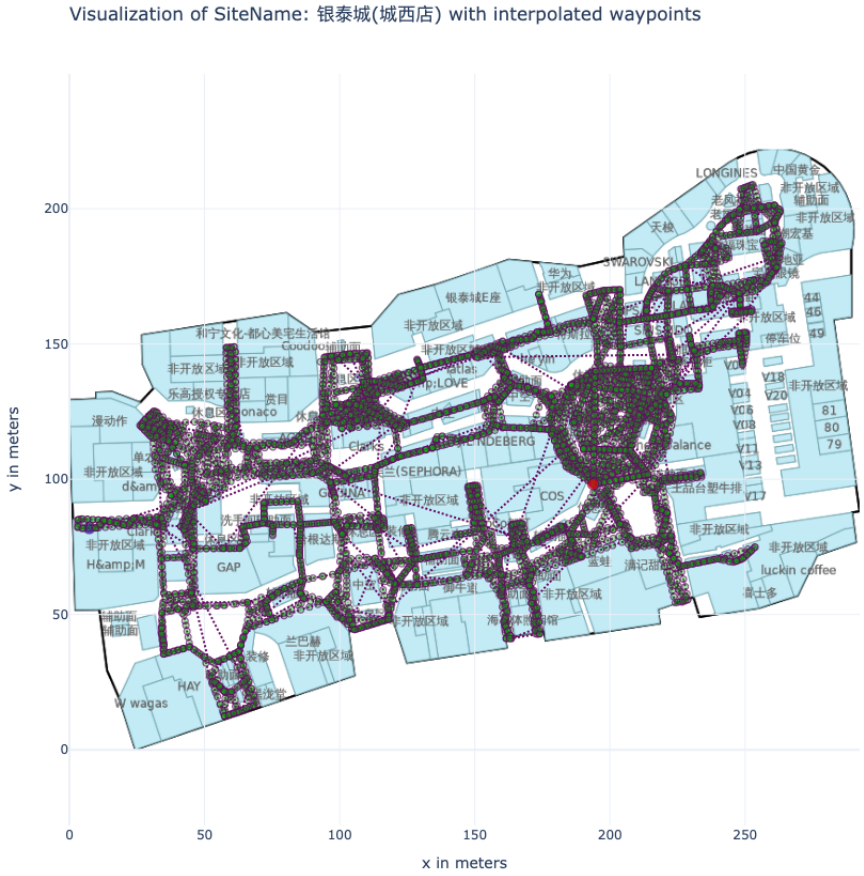

Figure 3.3: Visualization of the interpolated waypoints for SiteName: 银泰城(城西店)

4 Suitable Machine Learning Model

- As seen in the previous Chapter 3, many data for one floor
- As this is a time series, we want to use a machine-learning model suitable for time series

  kind of obvious, isnt it?

- As we want to predict the next BSSID out of many BSSIDs, we have multi-class classification

  why is it multi-class if you just want THE (single) next one??

  problem with supervised learning, as we know which BSSID is the next one
- Discussion of some pre-chosen models and decision for one which will be implemented in the next chapter

  indeed!

4.1 Classification Models

- Stated in Chapter 2, classification model in ML is a type of predictive model that categorizes incoming input data into specific classes

  but do we really have a classification here? in the narrow sense? classes of APs are given

- Prediction of next BSSID is a classification problem

  so, in the sense of classifying the sequence of recent APs into the proper class for "next AP" (or next APs)?

- therefore, interpret the problem as a classification problem

  is that the common approach? I suppose, but back it up by literature!

- For multivariate time series classification models, there are only a few models
  - MLPs [11]
  - RNNs such as LSTMs[5]
  - HMMs [18],

4.1.1 MLP

in headings, use full forms of the acronyms

MLP, also known as a feedforward artificial neural network, is a class of deep learning models primarily used for supervised learning tasks. An MLP consists of multiple layers of nodes in a directed graph, each fully connected to the next one. Each node in one layer is connected with certain weights to every node in the following layer. MLPs apply a series of transformations, typically nonlinear, to the input data using activation functions, such as the sigmoid or Rectified Linear Unit (ReLU), facilitating the model's ability to model complex patterns and dependencies in the data [4].

4.1.2 HMM

HMM is a statistical model that assumes the system being modeled is a Markov process with unobserved (hidden) states [19].

always a space before a cite

HMMs are mainly known for their application in temporal pattern recognition, such as speech and handwriting. They describe the probability of a sequence of observable data, which is assumed to result from a sequence of hidden states, each producing an observable output according to a particular probability distribution.

4.1.3 RNN

RNN is an artificial neural network well-suited to sequential data because of its intrinsic design. Unlike traditional feedforward neural networks, an RNN possesses loops in its topology, allowing information to persist over time. This unique characteristic enables the model to use its internal state (memory) to process sequences of inputs, making it ideally suited for tasks involving sequential data such as speech recognition, language modeling, and time series prediction[3].

4.1.3.1 LSTM

LSTM is a special kind of RNN, capable of learning long-term dependencies, which Hochreiter and Schmidhuber introduced in 1997[5]. LSTMs were designed to combat the "vanishing gradient" problem in traditional RNNs. This problem made it difficult for other neural networks to learn from data where relevant events occurred with significant gaps between them. The key to the ability of the LSTMs is its cell state and the accompanying gates (input, forget, and output gate), which regulate the flow of information in the network.

4.1.4 Discussion of Classification Models

As mentioned in Chapter 3, the floor analyzed there has 4795 BSSIDs. So we have 4795 classes for the classification problem. The selection of a suitable model for this task is even more critical. We will discuss the classification models according to the following aspects

properties?

: Temporal Dependency Handling, Capacity and Complexity, Multivariate Data, Flexibility and Integration, and Regularization and Overfitting.

Why these? Why are these aspects relevant for the problem at hand? Which are less important? Do you have a concrete expectation for these aspects BEFORE looking at the different techniques? Make it specific to YOUR problem, not just a generic discussion!

never use linebreaks in text. No colons at the end of headings. NEver use textbf as headings. Use descipriotn environments or simiarl.

Temporal Dependency Handling:

- MLPs have no loop [17], making them less suitable for time series data where temporal dependencies are crucial.
- While HMMs can handle temporal dependencies to some extent, they often struggle with longer sequences and multivariate data due to their Markovian assumption [19], which limits their memory to the most recent state.
- Standard RNNs were designed to handle temporal dependencies, but they suffer the vanishing gradient problem, making them less effective in capturing long-term dependencies compared to LSTMs [16].
- LSTMs, by design, are equipped to handle long-term temporal dependencies. Their unique cell state and gating mechanisms allow them to store, modify, and access information over extended periods, making them adept at capturing patterns from long sequences. [5]

Capacity and Complexity:

- MLPs can also scale their capacity by adding more hidden layers and units. They are capable of modeling complex relationships within data through their nonlinear activations.
- HMMs have limitations in handling the complexity of multi-class and multivariate problems due to their inherent Markovian assumptions and discrete state representations.
- traditional RNNs suffer from the vanishing gradient problem, especially in longer sequences, which limits their ability to capture long-term dependencies effectively. [16]
- With 4,795 classes, the model needs a considerable capacity to differentiate between the subtle differences in patterns that might exist among them. LSTMs, being deep learning models, can scale effectively in terms of capacity by adding more layers or units while still maintaining their ability to handle temporal data.

  that is the only really concrete, specific aspect, I think? anything I missed?

Multivariate Data:

- While MLPs can also handle multivariate data, they treat each feature and time step independently, often missing out on the interdependencies.
- HMMs are primarily designed for univariate data. Extending them to multivariate scenarios requires additional complexities and assumptions.
- RNNs and LSTMs can seamlessly handle multivariate time series data. Their recurrent nature allows them to effectively process each time step with multiple features.

Flexibility and Integration:

- MLPs are very flexible and can be used generally to learn a mapping from inputs to outputs. [2] As we want to learn, which BSSID is the next one, this may be a good fit.
- HMMs are primarily designed for capturing state transitions in sequential data and may not be suitable for tasks requiring the integration of spatial and temporal information. Their rigid assumptions about state transitions limit their flexibility in capturing complex patterns. [19]
- Traditional RNNs can capture short-term dependencies and are relatively simpler to integrate with other architectures due to their sequential nature.
- LSTMs can be easily integrated with other deep learning architectures, such as Convolutional Neural Networks (CNNs), to capture both temporal and spatial features. This flexibility is advantageous when dealing with complex and varied data sources.

Regularization and Overfitting:

- dropouts may be used to prevent overfitting for each model[20].
- RNNs are prone to vanishing gradient issues, which can increase overfitting in general. [16]

man hätte diese pure Beschreibung der verschiedenen Modelle nun auch gut nach Background an den Anfang legen können. aber ist ok, lassen wir hier.

In conclusion, while MLPs, HMMs, and traditional RNNs have their strengths and have been successful in many applications, they have problems with multivariate time series classification with many classes. This problem demands a model that can efficiently capture intricate temporal patterns, scale in capacity, and handle multivariate data. LSTMs, with their unique architecture and properties, can deal with this challenge, making them the preferred choice for this task and the selected model for our implementation.

well... this almost sounds like a foregone conclusion? Empahsis which aspects of the problem really caused this decision.

5 Implementation

obviously, this is a bit too brief in this form :-)

All the code for this implementation can be found in the GitHub repository [21].

5.1 Preprocessing

- Preprocessing with numpy and pandas
- Use data from Chapter 3 for preprocessing
- Explain what was done to the .txt files in preprocessing part (see [21] preparation.ipynb)

  yes, please! in more detail than in the earlier section!

- Explain preprocessing of model (see [21] lstm.ipynb)
- Load data from files of floor with most files
- Create a target variable (based on RSSIs of BSSIDs)
- Normalize the data.
- Create sequences of data based on window_size variable.
- Encode the target variable, which is a variable with 4795,

  4795 whats?

  where 1 means the class is the nearest AP and 0 means the class is not the nearest AP, which results in a one-hot encoding.

  why one-hot? ios this then still mutli-variate?

- Split the data into training and testing set. (80/20)

5.2 LSTM Training and Testing

- Use Keras library for implementation [13]
- Model: LSTM layer, Dense layer with softmax activation

  Sequences are of length window_size for each entry in the dataset

  Inputs are (window_size, number_of_features (which are 6 + number of BSSIDs)), see Fig. 5.1
- Generate predictions on the test set

- Get class with the highest probability as prediction
- Get top 3 predictions for the test set and check if the target variable is in the top 3 predictions.

why top 3 ? why not 2 or 5?

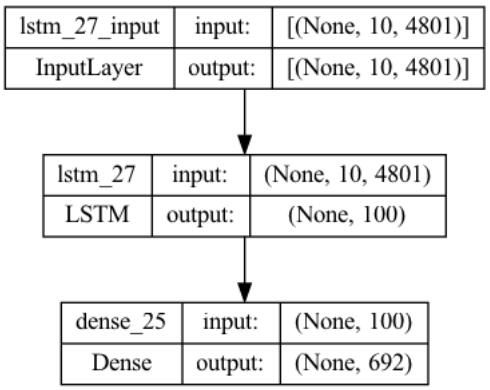

Figure 5.1: An example LSTM Network with Input, LSTM and Dense Layer with 4801 Features and window_size of 10.

uh... NIE pngs einbinden; immer Vektorformate!!

5.3 Tuning model and hyperparameters

- try out different hyperparameters also in combination

  Number of units in the LSTM layer = {100, 150, 200, 350, 500, 1000, 2000}

  batch size = {16, 32, 64}

  window size = {3, 5, 10, 20}

again: why these??? always give reasons for such decisions!

6 Evaluation

Variablen aus mehr als einem Buchstaben immer upright setzen. Es ist NUM, nicht $NUM = N \cdot U \cdot M$!

- Top k: predicted class is within the top k predictions
- calculation for top k:

$$\frac{\binom{NUM\_CLASSES-1}{k-1}}{\binom{NUM\_CLASSES}{k}} = \frac{k}{NUM\_CLASSES}$$

  what does that mean?

- for the floor with most files: NUM_CLASSES = 4795
- Probabilities to pick one random BSSID, and it is the right one in top 3, 5 or 10, see Fig. 6.1

  naja... aber Random ist jetzt kein sehr aussagefähiger Vergleichsfall, oder?

  was ist mit trivialen Vergelichsfällen? Z.B: ich nehme den letzten AP, davon die fünf nächsten, und suche mir zufällig einen aus? die drei nächsten? oder ... ? DAs wäre nahezu trivial zu realisieren und vermutlich auch gar nicht so schlecht???

- Accuracy of the model's prediction with a batch_size of 32, see Fig. 6.2,Fig. 6.3,Fig. 6.4 and Fig. 6.5
- Accuracy of the model's prediction with a batch_size of 16, see Fig. 6.6
- Accuracy of the model's prediction with a batch_size of 64, see Fig. 6.7

what is batch size?

seriously, PNGs???

- Comparison with random selection of classes: lstm always better than random selection

  well, one would STRONGLY hope so!

- TODO: describing the plots
- best performance: 71%, see Fig. 6.8

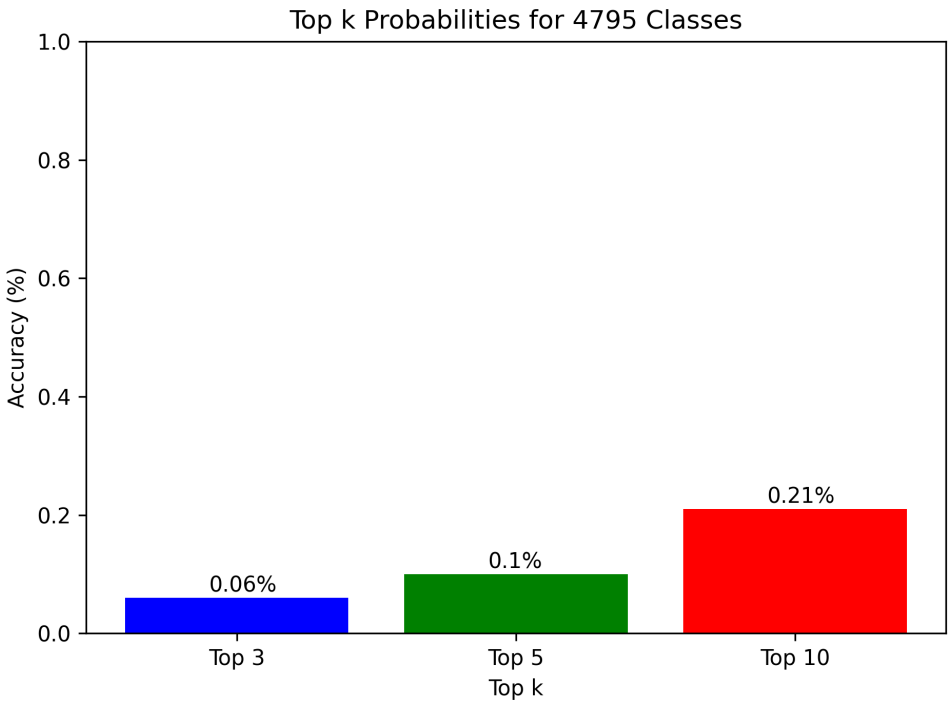

Figure 6.1: Probabilities that the predicted class falls within the top k randomly selected classes.

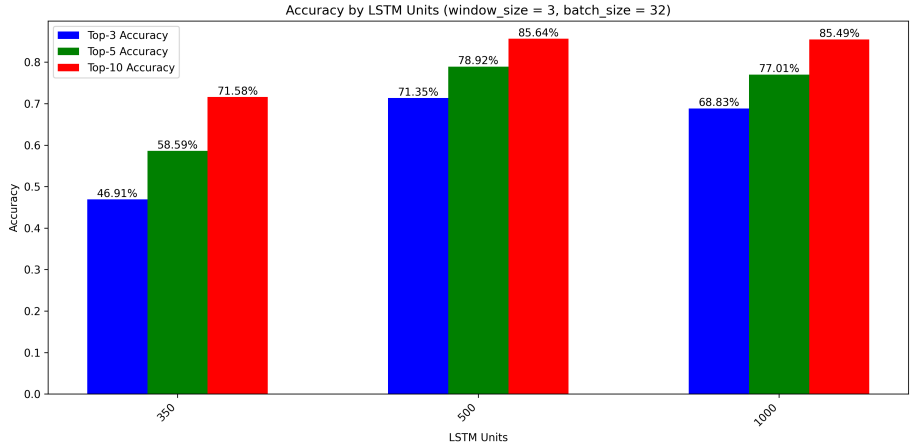

Figure 6.2: Accuracy of the model with window size of 3, batch size of 32 and 100, 500 and 1000 units in the LSTM layer.

- Reasons:

  data contains too many classes, with fewer classes to predict the model could have performed better

  discussion could have missed a better model

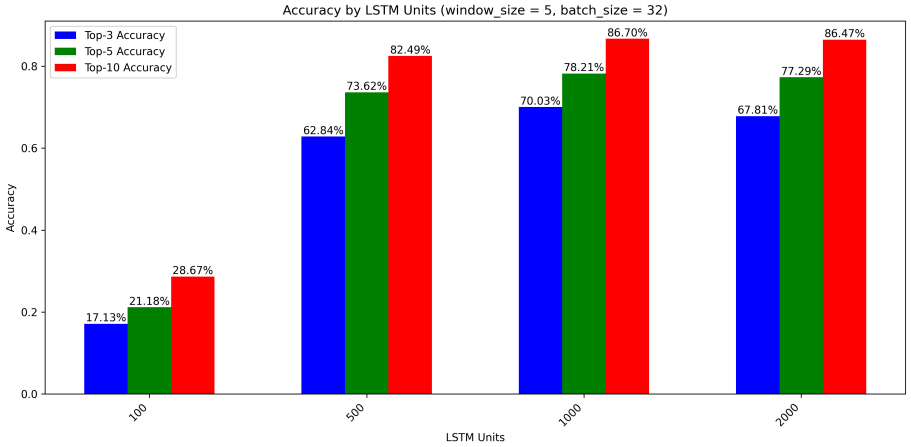

Figure 6.3: Accuracy of the model with window size of 5, batch size of 32 and 100, 500 and 1000 units in the LSTM layer.

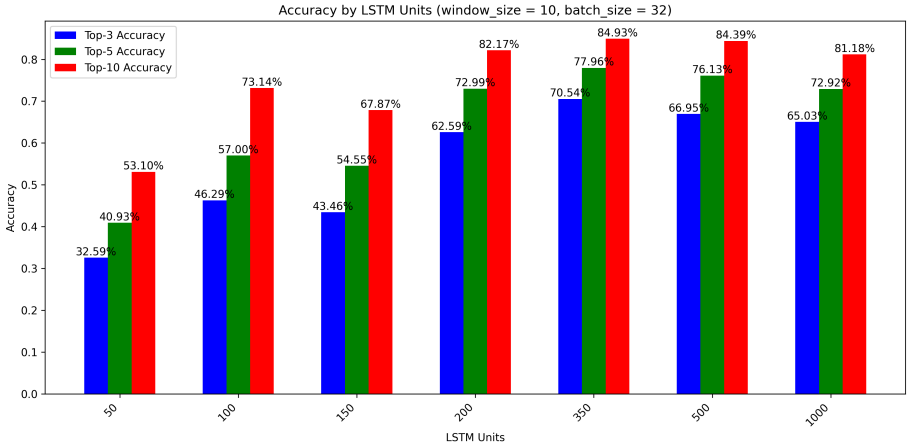

Figure 6.4: Accuracy of the model with window size of 10, batch size of 32 and 100, 500 and 1000 units in the LSTM layer.

model very simple, could be improved by using other layers in between LSTM and Dense layers

subitem is odd... usually, just net itemize environments; usually looks better

The results are what they are. THat's fine. But the comparison case of random choice really is too simplistic!! even some simple heuristic is likely to be much better than pure random choice?!?!? And should not be too much to implement?

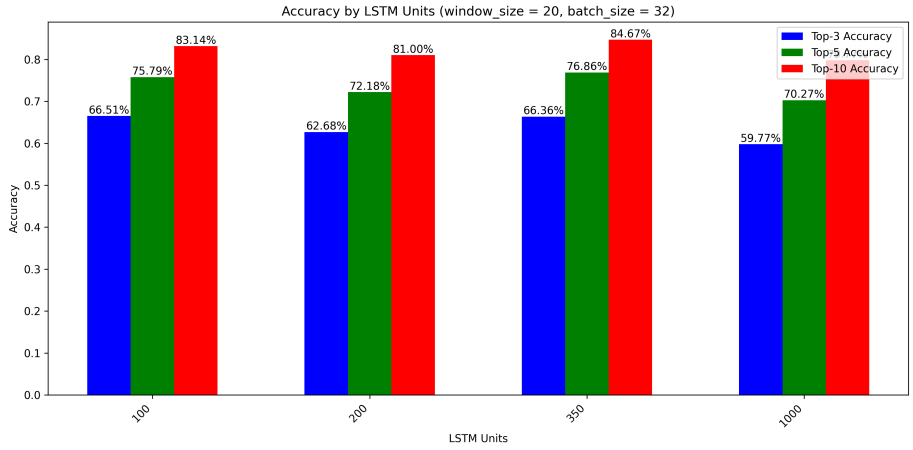

Figure 6.5: Accuracy of the model with window size of 20, batch size of 32 and 100, 500 and 1000 units in the LSTM layer.

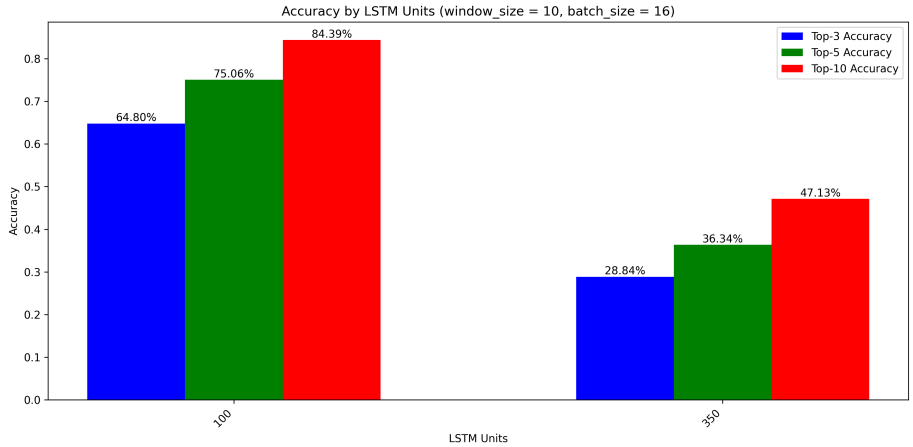

Figure 6.6: Accuracy of the model with window size of 10, batch size of 16 and 100, 500 and 1000 units in the LSTM layer.

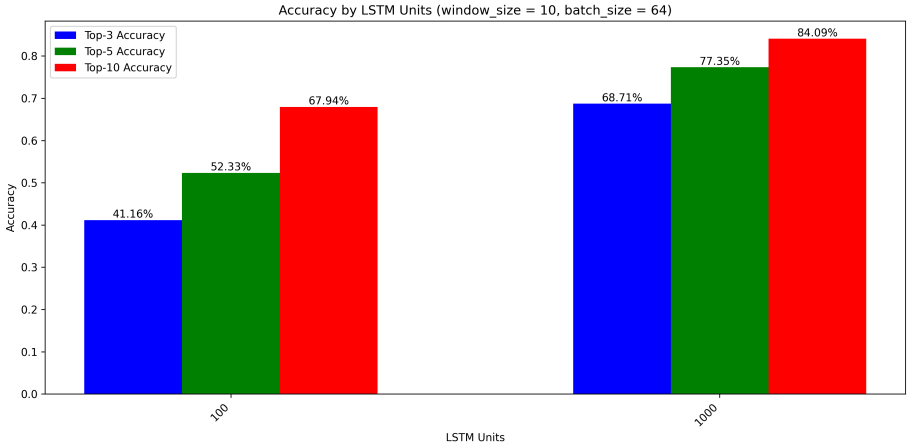

Figure 6.7: Accuracy of the model with window size of 10, batch size of 64 and 100, 500 and 1000 units in the LSTM layer.

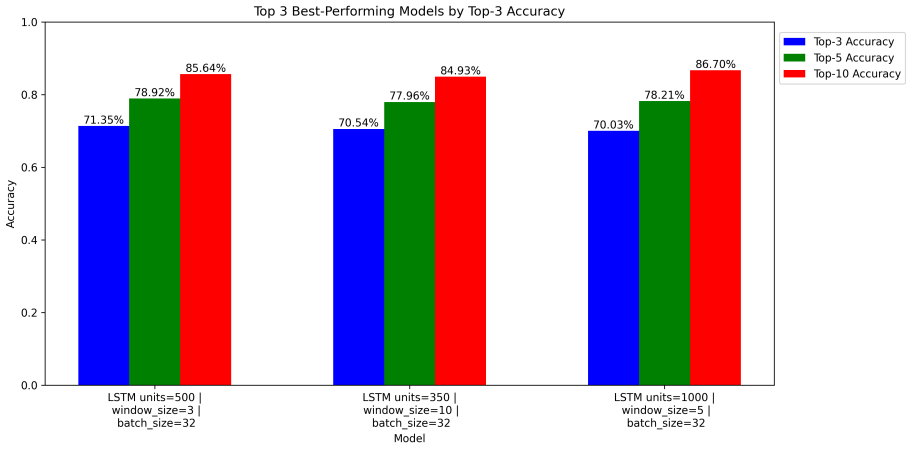

Figure 6.8: Accuracy of the three best performing models of the implementation

7 Conclusion

- indoor human movement prediction is a hard task with many classes
- although LSTM is the best choice for this task, the prediction accuracy is 70%
- with lesser classes,

  fewer... classes sind countable

  model could predict better, which could be a reason why ML model

  I am still strugglling to see where the classes come from, why there is no freedom of rediefining them, ...

- future work: use LSTM with fewer classes

  or generate data from mobile devices, so complete setup is known

  or generate data from APs to predict human movement

  could than be integrated in AP software like OpenWrt

Bibliography

[1] *Bishop - Pattern Recognition and Machine Learning.pdf*. URL: https://docs.google.com/viewer?a=v&pid=sites&srcid=aWFtYW5kaS5ldXxpc2N8Z3g6MjViZDk1NGI1NjQzOWZiYQ (visited on July 21, 2023).

[2] Jason Brownlee. *When to Use MLP, CNN, and RNN Neural Networks*. URL: https://machinelearningmastery.com/when-to-use-mlp-cnn-and-rnn-neural-networks/ (visited on Aug. 23, 2023).

[3] Jeffrey L. Elman. "Finding Structure in Time". en. In: *Cognitive Science* 14.2 (Mar. 1990), pages 179–211. ISSN: 03640213. DOI: 10.1207/s15516709cog1402_1. URL: http://doi.wiley.com/10.1207/s15516709cog1402_1 (visited on Aug. 6, 2023).

[4] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. eng. OCLC: 987005922. Cambridge, Massachusetts: The MIT Press, 2016. ISBN: 9780262337434.

[5] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: (1997). URL: https://papers.baulab.info/Hochreiter-1997.pdf.

[6] "IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 2: Fast Basic Service Set (BSS) Transition". In: *IEEE Std 802.11r-2008 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008)* (July 2008). Conference Name: IEEE Std 802.11r-2008 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008), pages 1–126. DOI: 10.1109/IEEESTD.2008.4573292.

[7] "IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 11: Wireless LAN Medium Access Control (MAC)and Physical Layer (PHY) Specifications Amendment 1: Radio Resource Measurement of Wireless LANs". In: *IEEE Std 802.11k-2008 (Amendment to IEEE Std 802.11-2007)* (June 2008). Conference Name: IEEE Std 802.11k-2008 (Amendment to IEEE Std 802.11-2007), pages 1–244. DOI: 10.1109/IEEESTD.2008.4544755.

[8] *Indoor Location & Navigation | Kaggle*. https://www.kaggle.com/competitions/indoor-location-navigation. (Visited on July 11, 2023).

[9] *indoor-location-navigation-20*. URL: https://github.com/location-competition/indoor-location-competition-20 (visited on July 31, 2023).

[10] *Indoor Navigation: Complete Data Understanding*. https://kaggle.com/code/andradaolteanu/indoor-navigation-complete-data-understanding. (Visited on Apr. 25, 2023).

[11] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. "Deep learning for time series classification: a review". en. In: *Data Mining and Knowledge Discovery* 33.4 (July 2019), pages 917–963. ISSN: 1384-5810, 1573-756X. DOI: 10.1007/s10618-019-00619-1. URL: https://link.springer.com/10.1007/s10618-019-00619-1 (visited on Aug. 6, 2023).

[12] *Kaggle: Your Home for Data Science*. https://www.kaggle.com/. (Visited on July 23, 2023).

[13] *Keras: The high-level API for TensorFlow*. URL: https://www.tensorflow.org/guide/keras (visited on Aug. 18, 2023).

[14] Joos Korstanje. *How to Select a Model For Your Time Series Prediction Task*. URL: https://neptune.ai/blog/select-model-for-time-series-prediction-task (visited on Aug. 23, 2023).

[15] Tom O'Malley, Elie Bursztein, James Long, François Chollet, Haifeng Jin, Luca Invernizzi, et al. *KerasTuner*. https://github.com/keras-team/keras-tuner. 2019. (Visited on Aug. 21, 2023).

[16] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. *On the Difficulty of Training Recurrent Neural Networks*. Feb. 2013. arXiv: 1211.5063 [cs]. (Visited on Aug. 17, 2023).

[17] Marius-Constantin Popescu, Valentina Balas, Liliana Perescu-Popescu, and Nikos Mastorakis. "Multilayer perceptron and neural networks". In: *WSEAS Transactions on Circuits and Systems* 8 (July 2009).

[18] Pratap S. Prasad and Prathima Agrawal. "Movement Prediction in Wireless Networks Using Mobility Traces". In: *2010 7th IEEE Consumer Communications and Networking Conference*. Jan. 2010, pages 1–5. DOI: 10.1109/CCNC.2010.5421613.

[19] L.R. Rabiner. "A tutorial on hidden Markov models and selected applications in speech recognition". In: *Proceedings of the IEEE* 77.2 (Feb. 1989), pages 257–286. ISSN: 00189219. DOI: 10.1109/5.18626. URL: http://ieeexplore.ieee.org/document/18626/ (visited on Aug. 6, 2023).

[20] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research* 15.56 (2014), pages 1929–1958. URL: http://jmlr.org/papers/v15/srivastava14a.html.

[21] Lina Wilske. *GitHub Repository for bachelor thesis*. URL: https://github.com/linaScience/ba-implementation.

Acronyms

AP	Access Point
BSSID	Basic Service Set Identifier
HMM	Hidden Markov Model
LSTM	Long Short-Term Memory
MAC	Media Access Control
MajorID	Major Identifier
MinorID	Minor Identifier
ML	Machine Learning
MLP	Multilayer Perceptron
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
RSSI	Received Signal Strength Indication
SSID	Service Set Identifier
TxPower	Transmission Power
UUID	Universally Unique Identifier
Wi-Fi	Wireless Fidelity

Zusammenfassung

To-do: translate english abstract to german

Eidesstattliche Erklärung

Hiermit versichere ich, dass meine Bachelor's thesis "Machine Learning-Based User Movement Prediction in Layer 2 Networks" ("Vorhersage von Benutzerbewegungen in Layer 2 Netzen basierend auf Maschinellem Lernen") selbstständig verfasst wurde und dass keine anderen Quellen und Hilfsmittel als die angegebenen benutzt wurden. Diese Aussage trifft auch für alle Implementierungen und Dokumentationen im Rahmen dieses Projektes zu.

Potsdam, den 24. August 2023,

_____

(Lina Wilske)