Enhanced Indoor Positioning:

Comparing LSTM Neural Networks with Traditional Regression Models for Accurate RSSI-Based Localization

Jhih-ci Chao & Sheroz Shaikh

Table of contents

1 Introduction

1

Tech Advancements

Confluence of **Mobile Devices** & **Smart Wearables**

2

Ubiquitous Computing

Revolutionizing Ambient Intelligence & Smart Environments

3

A Challenge

Indoor Localization with **Precision** and **Adaptability**

Background

Geo-Magnetic field and WLAN dataset for indoor localisation from wristband and smartphone

Donated on 1/9/2017

A multisource and multivariate dataset for indoor localisation methods based on WLAN and Geo-Magnetic ï¬eld ï¬ngerprinting

Dataset Characteristics	Subject Area	Associated Tasks
Multivariate, Sequential, Time-Series	Computer Science	Classification, Regression, Clustering
Feature Type	**# Instances**	**# Features**
Integer, Real	153540	25

Dataset Information

Additional Information

Indoor localisation is a key topic for the Ambient Intelligence (AmI) research community.

In this scenarios, recent advancements in wearable technologies, particularly smartwatches with built-in sensors, and personal devices, such as smartphones, are being seen as the breakthrough for making concrete the envisioned Smart Environment (SE) paradigm.

In particular, scenarios devoted to indoor localization represent a key challenge to be addressed. Many works try to solve the indoor localization issue, but the lack of a common dataset or frameworks to compare and evaluate solutions represent a big barrier to be overcome in the ï¬eld. The unavailability and uncertainty of public datasets hinders the possibility to compare different indoor localization algorithms. This constitutes the main motivation of the proposed dataset described herein.

We collected Wi-Fi and geo-magnetic ï¬eld ï¬ngerprints, together with inertial sensor data during two campaigns performed in the same environment. Retrieving syncronized data from a smartwatch and a smartphone worn by users at the purpose of create and present a public available dataset is the goal of this work.

SHOW LESS

DOWNLOAD

CITE

0 citations

1859 views

Creators

Paolo Barsocchi

Antonino Crivello

Davide Rosa

Filippo Palumbo

DOI

10.24432/C5DW43

License

Dataset Breakdown

Sensor

Sensor data from smartphones/watches

Timestamps

Timestamps of user location for movement tracking.

Wi-Fi

Signal strength measurements from WAPs.

Points Mapping

Mapping file linking place IDs to local X-Y coordinates

Sensor

Sensor data from smartphones/watches

Smartphone/smartwatch Sensors (*measure1(2)_smartphone(watch)_sens.csv*): Provide sensor data collected from the smartphone, with each row corresponding to a sensor reading at a specific timestamp.

- timestamp: Unix timestamp of when the sensor reading was taken.
- AccelerationX/Y/Z: Acceleration readings along the X, Y, and Z axes, respectively.
- MagneticFieldX/Y/Z: Magnetic field strength along the X, Y, and Z axes, respectively.
- Z-AxisAngle(Azimuth), X-AxisAngle(Pitch), Y-AxisAngle(Roll): Angular position of the device in degrees.
- GyroX/Y/Z: Gyroscope readings along the X, Y, and Z axes, respectively, indicating the device's rotational movement. Only for smartwatches.

Wi-Fi

Signal strength
measurements from WAPs

Smartphone Wi-Fi (*measure1(2)_smartphone_wifi.csv*): These files detail the Wi-Fi signal strength readings from various wireless access points (WAPs) at different locations.

- The first column represents the PlaceId, which corresponds to the ID in the Points Mapping file.
- The subsequent 127 columns represent the Received Signal Strength Indicator (RSSI) levels for different WAPs. These are typically in dBm (decibels relative to a milliwatt), with -100 dBm indicating no detection of a WAP.

Timestamps

Timestamps of user location
for movement tracking

Timestamp ID (*measure1(2)_timestamp_id.csv*): Contain timestamps marking the user's arrival and departure times at specific locations, associated with Place IDs.

- arrival: Unix timestamp indicating when the user arrived at a location.
- departure: Unix timestamp indicating when the user left the location.
- place_id: The unique identifier of the place corresponding to the ID in the Points Mapping file, ranging from 0 to 324.

Points Mapping

Mapping file linking place IDs
to local X-Y coordinates

Points Mapping (*PointsMapping.ods*): Maps physical locations in a space to a coordinate system.

- ID: Unique identifier for a specific location within the indoor map.
- X: The x-coordinate of the location in the local coordinate system.
- Y: The y-coordinate of the location in the local coordinate system.

2

EDA

Dataset Breakdown

Sensor

Sensor data from smartphones/watches

Timestamps

Timestamps of user location for movement tracking.

Wi-Fi

Signal strength measurements from WAPs.

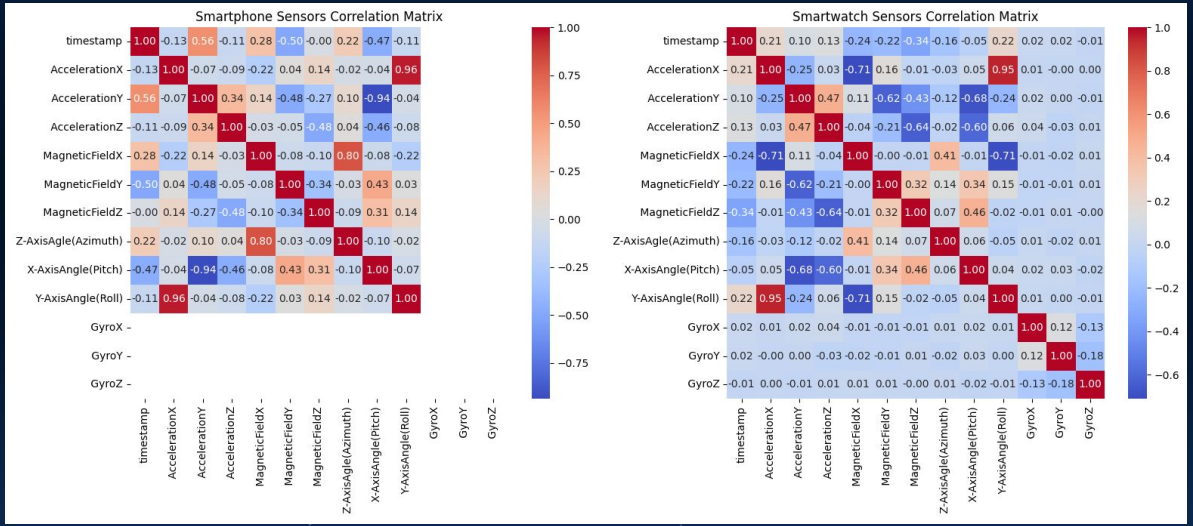
Points Mapping

Mapping file linking place IDs to local X-Y coordinates

Sensor

Sensor data from smartphones/watches

Correlation Matrix



Sensor

Sensor data from smartphones/watches

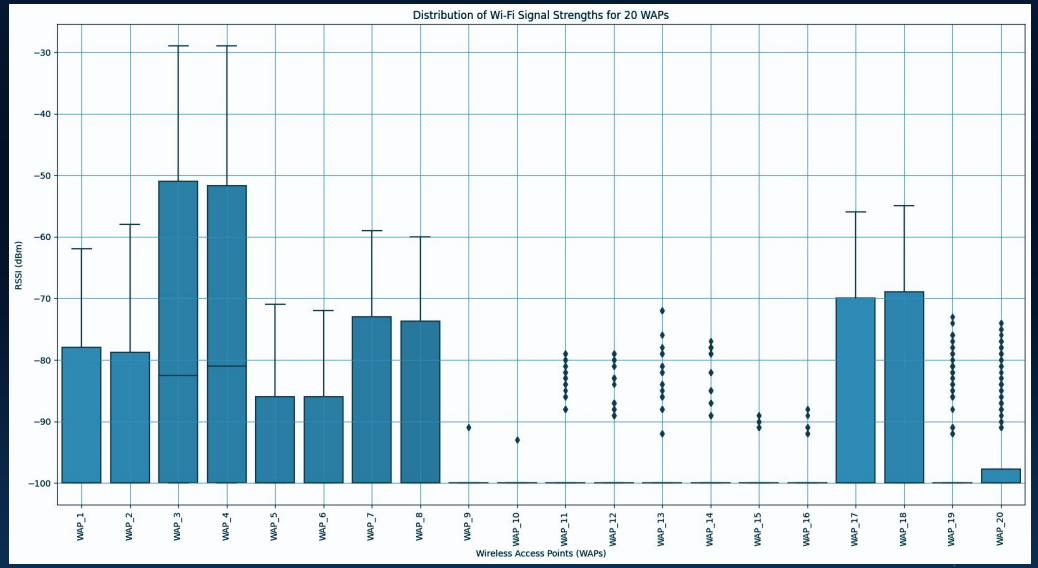
Key Correlations from Heatmap

- **Acceleration axes show some degree of correlation, expected due to the movement in 3D space.**
- **Magnetic field readings show varying degrees of correlation; they can be influenced by the orientation of the device relative to the Earth's magnetic field.**
- **Gyroscope readings (GyroX, GyroY, GyroZ) show low correlation with other sensors, which is typical since they measure rotation, not linear acceleration or magnetic fields.**

Wi-Fi

Signal strength measurements from WAPs

Boxplot of a subset of 20 WAPs.



Wi-Fi

Signal strength
measurements from WAPs

The Wi-Fi data consists of rows representing different Place IDs with 127 columns of Received Signal Strength Indicator (RSSI) values for various Wireless Access Points (WAPs). A value of -100 dBm indicates a WAP that was not detected.

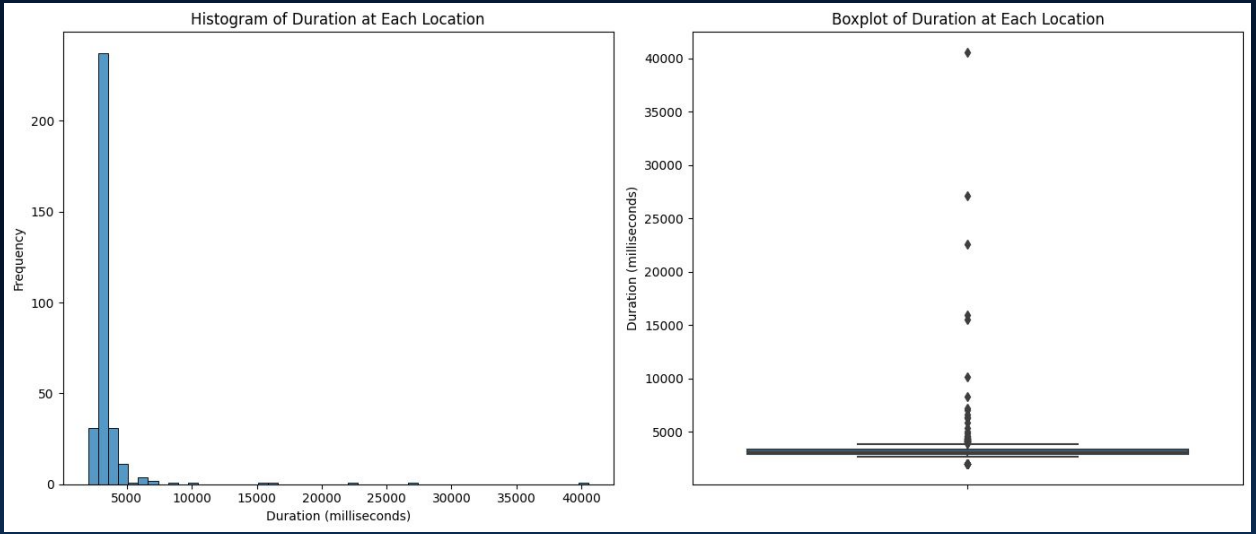
Statistics Summary:

- The RSSI values for the WAPs show a lot of variability. Many WAPs have a 25th percentile value of -100 dBm, indicating a high prevalence of non-detections.
- The mean values suggest that for most WAPs, the signal strength is weak (closer to -100 dBm)
- Signal strength can vary significantly even within the same WAP..

Timestamps

Timestamps of user location for movement tracking

Visualization of Durations



Timestamps

Timestamps of user location
for movement tracking

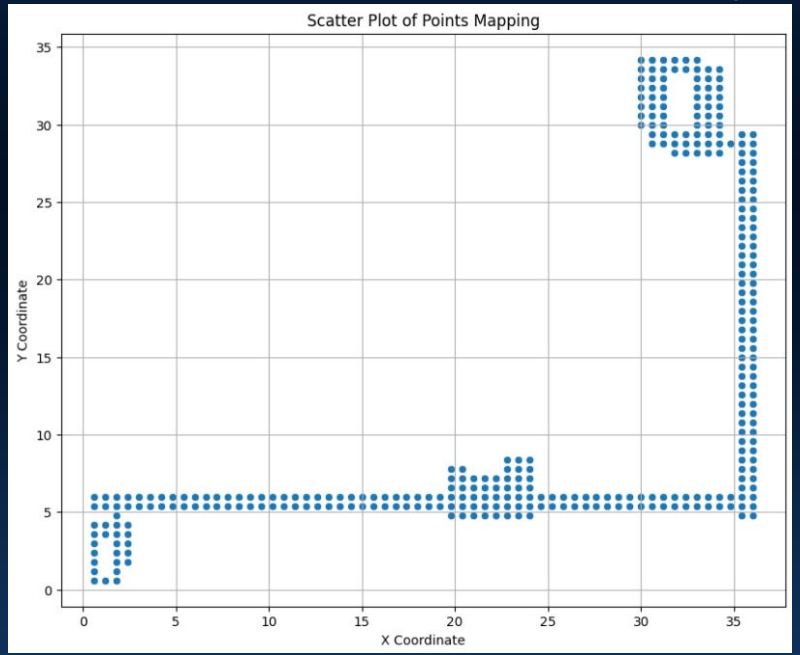
The duration data reveals the following:

- **The average duration of stay at a location is around 3559 milliseconds, with a standard deviation of approximately 2964 milliseconds.**
- **The minimum duration recorded is 2000 milliseconds, and the maximum is quite high at 40536 milliseconds, suggesting some longer stays or potential outliers.**
- **The majority of durations fall within a narrower range, as indicated by the 25th to 75th percentile values (2895 to 3291 milliseconds).**

Points Mapping

Mapping file linking place IDs to local X-Y coordinates

Points Mapping Plot



Points Mapping

Mapping file linking place IDs
to local X-Y coordinates

The scatter plot above visualizes the distribution of these points in the local coordinate system, showing how they are spread out.

- There don't appear to be any anomalies at first glance, as the points seem uniformly distributed without any obvious outliers.
- There's a visible pattern of higher density in certain regions, which could correlate to important areas within the indoor environment, such as entry points, rooms, or corridors.

3

Problem Definition

Observations

WAP Signal Strengths

Observed Issues

Outliers

Many signal outliers or non-detections were included.

Fluctuations

Signal strengths varied between and even within WAPs.

Current Limitations

Wi-Fi signal strength is affected by factors such as moving objects and reconfiguration of indoor spaces.

This variability is a significant source of error in indoor localization efforts and points to the need for a predictive model that can intelligently adapt to these temporal changes.

Problem Statement

The current localization models, which often assume static signal conditions, struggle to accommodate this variability, leading to reduced accuracy in real-time applications.

Significance

Time-Dependent RSSI Prediction

Important for creating responsive environments for enhanced user experience.

Localization Applications

Critical in scenarios like emergency evacuation and aiding the visually impaired.

Space Optimization

Potential in improving commercial and industrial space utilization.

AmI Vision

Theoretical and practical implications for the future of AmI.

4

Data Preprocessing

Data Merging

Merged sensor data, Wi-Fi signal strength data, and mapped them to the corresponding location coordinates.

Feature Generation

Time Features: Extract hour, minute, day of the week, and month from timestamps to capture time-related signal strength fluctuations.

Fourier Transforms: Identify frequency components in sensor data, useful for detecting periodic RSSI fluctuations.

Radians and Trigonometric Features: Convert angles to radians, create sine and cosine features for effective orientation capture, crucial for models like LSTMs.

RSSI Statistics: Compute mean and variance of RSSI values for all WAPs, summarizing signal strength and variability to better understand signal dynamics.

Interaction Features: Multiply sensor readings to capture interactions between different sensor inputs, as relationships between sensor inputs may not be simply additive.

Distance to Origin: Calculate Euclidean distance to a reference point to assess correlation with signal strength patterns.

Feature Selection

Data Sampling: A 10% random sample of the dataset was taken, using sample(frac=0.1, random_state=1). This ensures an effective and representative subset of the data.

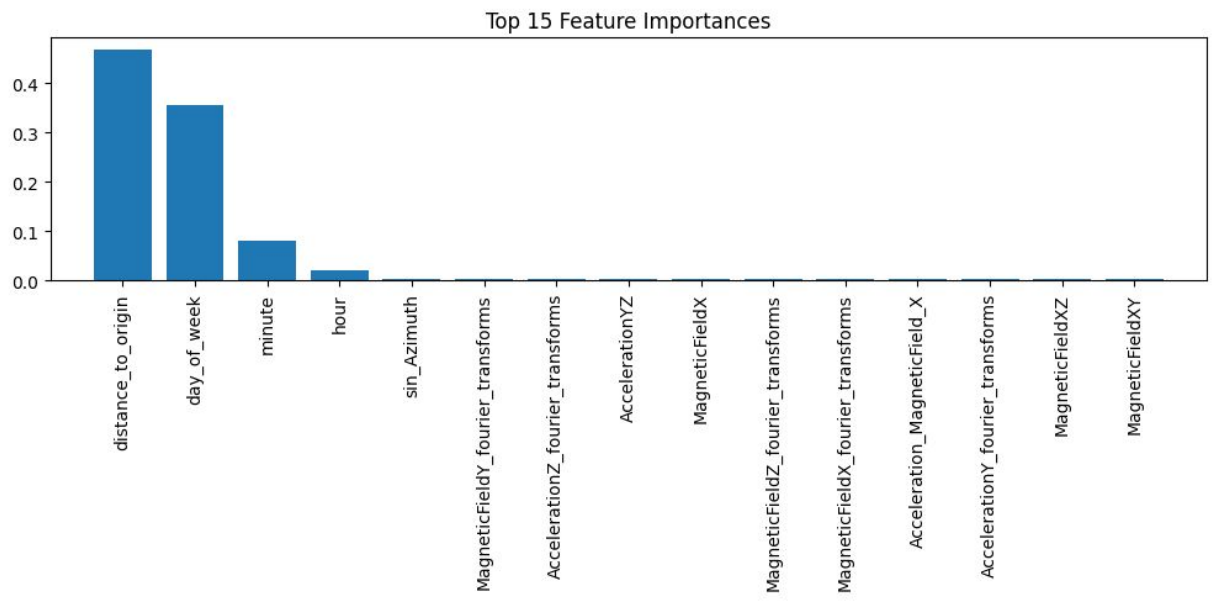
RSSI Column Selection: A subset of 10 RSSI columns was randomly selected from the dataset for the analysis for a diverse yet manageable set of signal strength features.

Feature Importance Accumulation: A dictionary was initialized to accumulate the importances of features across all selected RSSI columns.

Iterative Analysis for Each RSSI Column:

- **Data Preparation:** Non-predictor columns (other RSSI columns, timestamps, wifi_tag, and location identifiers) were excluded.
- **Train-Test Split:** 80-20 train-test split to validate model performance on unseen data.
- **Random Forest Model:** An RF regressor with 100 trees was trained, leveraging its capability to handle numerous features and capture complex relationships.
- **Importance Aggregation:** Feature importances were aggregated across all analyzed RSSI columns to gain a comprehensive understanding of each feature relevance.

Feature Selection



Feature Normalization

Selection of Features for Normalization

- Features selected for normalization exclude categorical and location-specific identifiers that do not require scaling.
- Ensured that only relevant numerical features are included for normalization.

Application of Min-Max Scaling

- The MinMaxScaler from sklearn.preprocessing module is employed, with a feature range set to (0, 1).
- The fit_transform is applied to the selected features of the data frame, ensuring that all relevant numerical data are scaled uniformly.

5

Model Implementation

Goal: Comparison

Traditional

Traditional Regression Models

LSTM

Time-Series-Specific Model

Traditional Regression Models

Linear Regression: Simple and interpretable baseline model.

Ridge Regression: Adds L2 regularization to address multicollinearity.

Lasso Regression: Uses L1 regularization for sparse feature selection.

Random Forest Regression: Decision tree-based ensemble method for robust predictions.

Gradient Boosting Regression: Enhances decision trees for better performance.

Elastic Net Regression: Combines L1 and L2 regularization, adjustable with l1_ratio.

Bayesian Ridge Regression: Probabilistic approach to linear regression, with flexible hyperparameters.

Decision Tree Regression: Non-parametric, splits features for complex data patterns.

K-Nearest Neighbors Regression: Predicts based on the average of k-nearest neighbors; 'k' and distance metric are key.

Huber Regressor: Merges mean squared and absolute error, less sensitive to outliers with an epsilon parameter.

Traditional Regression Models - Optimization

Feature Engineering: We considered feature importance from tree-based models to select relevant features, eliminating noise and improving model efficiency.

Cross-Validation: Utilized cross-validation to assess the models' performance across different subsets of the training data, ensuring robustness and generalization.

Hyperparameter Tuning: Employed random grid search to find optimal hyperparameters for selected models, fine-tuning them for improved performance.

LSTM – Why?

Long Short-Term Memory is a type of Recurrent Neural Network (RNN) specialized in processing sequences and time-series data.

- First proposed by Sepp Hochreiter and Jürgen Schmidhuber in 1997
- Addresses the vanishing gradient problem of neural networks, particularly those using backpropagation and gradient-based learning methods.
- Excel at learning from data where current events are influenced by long-past events.

Key Features

- Memory Cells: Capable of maintaining information in 'memory' for long periods.
- Gates (Input, Output, Forget): Control the flow of information, allowing the network to selectively remember or forget patterns.

LSTM – Autocorrelation Analysis

Autocorrelation measures how a data series correlates with itself over different time lags.

- Indicates how much current data points resemble past values.
- Crucial for understanding patterns and predictability in time series data.
- Helps analyze the stability and consistency of signal strength over time.

Definition: The autocorrelation of a time series X_t at lag k is defined as the correlation between values of the series at different times, as measured by the lag between them.

Mathematical Formula: The autocorrelation function $R(k)$ for a time series X_t is given by:

$$R(k) = \frac{E[(X_t - \mu)(X_{t-k} - \mu)]}{\sigma^2}$$

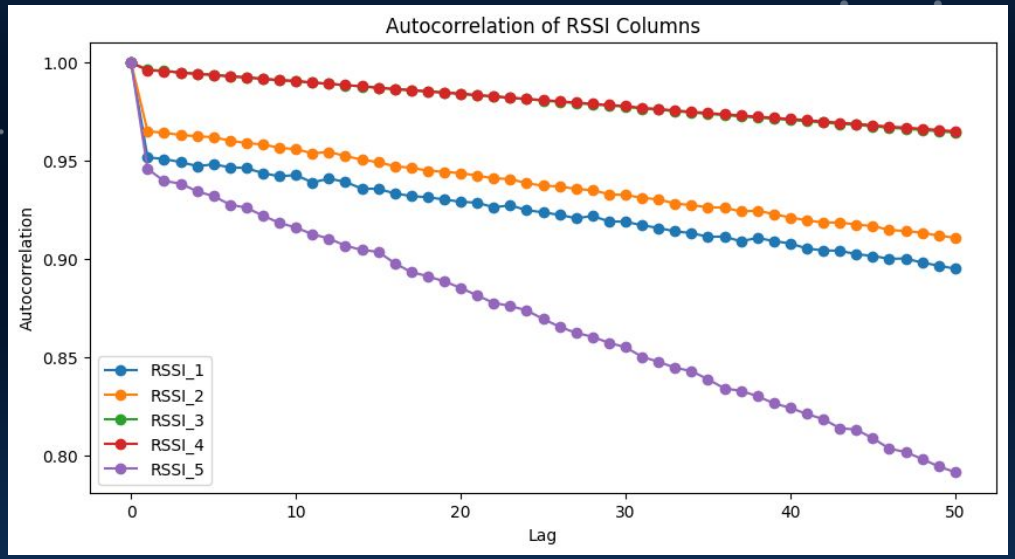
where:

- E is the expected value operator.
- μ is the mean of the time series.
- σ^2 is the variance of the time series.
- k is the lag.
- X_t and X_{t-k} are the values of the series at times t and $t - k$, respectively.

Normalization: The autocorrelation value at lag 0 (i.e., $R(0)$) is always 1 because it's the correlation of the series with itself. Autocorrelation values typically range between -1 and 1, where:

- 1 indicates perfect positive correlation.
- -1 indicates perfect negative correlation.
- 0 indicates no correlation.

LSTM – Autocorrelation Analysis



`acf` from `statsmodels.tsa.stattools`, on each of the 5 chosen RSSI columns.

LSTM – Autocorrelation Analysis / Model Configuration

Model Configurations

Model takes long to run, we weren't able to test many configurations

- **Sequence Length:** number of consecutive data points from the past that the model uses to make a prediction
  - Set as **12**
  - Autocorrelation remained strong after 50 lags, indicating high influence of past values.
  - SL of 12 allows the model to look at a sufficiently long history of data points.
- **Prediction Length:** number of future data points the model is trained to predict.
  - Set as **5**
  - Slow decline of autocorrelation observed.
  - Balances immediate forecasting and accuracy.
- **Feature Columns:** 'distance_to_origin', 'day_of_week', 'minute', and 'hour'

LSTM – Implementation (Architecture & Compilation)

LSTM Architecture:

- **Two LSTM Layers:** Each with 50 units, designed to process sequence data.
- **First LSTM Layer:** Returns sequences, maintaining temporal relationships between data points.
- **Second LSTM Layer:** Compresses these sequences into a single context vector, capturing essential features.
- **Dense Layer:** Reshapes the output to align with the required prediction length and feature dimensions.

Model Compilation:

- **Adam Optimizer:** A widely-used optimizer that adjusts the network weights iteratively based on training data. Known for its efficiency in handling large datasets and its adaptive learning rate.
- **Loss Function - Mean Squared Error (MSE):** Measures the average of the squares of errors, i.e., the difference between actual and predicted values. Ideal for regression tasks where you predict continuous values.

LSTM – Implementation (Training & Optimization)

Training Process:

- **Epochs (8):** number of times the dataset is passed forward and backward through the neural network.
- **Batch Size (64):** Number of samples processed before the model is updated.

Optimization:

- Rough estimate of sequence and prediction lengths using the autocorrelation analysis.
- Test a few different values of the parameters and model configuration.

6 Evaluation

Traditional Regression Models - Performance

Metrics used: MSE, RMSE, NRMSE, R^2, MAE, MAPE, MBD, MASE, EVS

Below is a high-level overview of the insights gathered from the report:

Performance Comparison: We observed that the Random Forest and Gradient Boosting Regression models outperformed linear models (Linear, Ridge, Lasso) in terms of MSE, RMSE, and R2 Score.

- **Impact of Optimization:** Optimization techniques, including feature engineering and hyperparameter tuning, positively influenced model performance. The optimized models exhibited improved accuracy and robustness.
- **Model Robustness:** Random Forest and Gradient Boosting demonstrated better resilience to variations in the dataset, contributing to their superior performance.
- **Feature Importance:** Models with feature importance considerations showed enhanced predictive power by focusing on relevant features.

Result: Random Forest and Gradient Boosting Regression are promising models for predicting RSSI values in our specific context.

LSTM – Performance

Key Metrics:

- **Mean Absolute Error (MAE) - 0.0023**
- **Mean Squared Error (MSE) - 0.0001**
- **Root Mean Squared Error (RMSE) - 0.0124:** The square root of MSE, offering a more interpretable scale. Low RMSE indicates good model performance.
- **R-squared (R^2) - 0.9994:** Nearly perfect.
- **Mean Absolute Scaled Error (MASE) - 0.0312:** Compares the model's error to that of a naïve benchmark. Low MASE indicates superior forecasting accuracy.
- **Explained Variance Score (EVS) - 0.9994:** Similar to R^2, measures the proportion of variance in the dependent variable predicted from the independent variable(s).

Conclusion

Model Comparison:

- **LSTM Superiority:** Demonstrates advanced capabilities in handling complex time-series data, outperforming traditional models in key areas like MAE, MSE, RMSE, R2, and EVS.
- **Significance:** Highlights LSTM's strength in capturing temporal sequences and dependencies, crucial for accurate time-series forecasting.

Possible Future Works:

- **Real-Time Data Processing:** Implement a pipeline for real-time data analysis and prediction, enhancing the model's applicability in dynamic environments.
- **Comparative Analysis with Other Models:** Benchmark against other advanced models like GRU or Transformer-based architectures for a comprehensive performance evaluation.
- **Deployment and Scalability:** Assess model's scalability and deployment strategies in real-world scenarios, including cloud integration and edge computing.
- **Interpretable AI Methods:** Apply techniques for model interpretability to provide insights into how and why predictions are made, enhancing trust and understanding.

Thanks!

CREDITS: This presentation template was created by **Slidesgo**, and includes icons by **Flaticon** and infographics & images by **Freepik**

Visuals generated by DALL-E using ChatGPT