TITLE

Machine learning approaches based on local magnetic field strength recorded by magneto-inertial measurement units to discern indoor from outdoor environments during activities of daily living

AUTHORS

V. Marcianò1,\*, S. Bertuletti2, N. Ireson3, F. Ciravegna3, A. Bevilacqua4, G. Ifrim4, B. Caulfield4,5, T. Bonci6,7, C. Mazzà6,7, L. Alcock8, S. Del Din8, E. Gazit9, J.M. Hausdorff9, A. Cereatti1 for the Mobilise-D consortium

AFFILIATIONS

1. Department of Electronics and Telecommunications, Politecnico di Torino, Turin, Italy
2. Department of Biomedical Sciences, University of Sassari, Sassari, Italy
3. Department of Computer Science, University of Sheffield, Sheffield, UK
4. Insight Centre for Data Analytics, University College Dublin, Dublin, Ireland
5. School of Public Health, Physiotherapy and Sports Science, University College Dublin, Dublin, Ireland
6. Department of Mechanical Engineering, The University of Sheffield, Sheffield, UK
7. Insigneo Institute for in silico Medicine, The University of Sheffield, Sheffield, UK
8. Translational and Clinical Research Institute, Faculty of Medical Sciences, Newcastle University, Newcastle upon Tyne, UK
9. Center for the Study of Movement, Cognition and Mobility, Neurological Institute, Tel Aviv Sourasky Medical Center, Tel Aviv, Israel

\* Corresponding author

KEYWORDS

indoor-outdoor detection, machine learning, time series classification, wearable sensors, magneto-inertial measurement units, magnetic field.

ABSTRACT (150 to 250 words)

1. INTRODUCTION
2. MATERIALS AND METHODS
   1. Data collection

A sample of 20 participants were recruited in three sites: The Newcastle upon Tyne Hospitals NHS Foundation Trust, UK and Sheffield Teaching Hospitals NHS Foundation Trust, UK (ethics approval granted by London – Bloomsbury Research Ethics committee, 19/LO/1507), and Tel Aviv Sourasky Medical Center, Israel (ethics approval granted by the Helsinki Committee, Tel Aviv Sourasky Medical Center, Tel Aviv, Israel, 0551-19TLV). Each participant gave written informed consent to take part in the studies.

The full description of the experimental setup and protocol can be found in [1]. Each participant was monitored for 2.5 hours in his/her habitual environment (e.g., home, work, etc.) and was asked to perform some specific tasks such as outdoor walking, walking up and down a slope and stairs, and moving from one room to another. Each participant wore a magneto-inertial measurement unit (MIMU) [2] on each foot (left foot: LF; right foot: RF), lower-back (LB), and non-dominant wrist (WR) (Figure 1). Each MIMU integrates a triaxial accelerometer (full scale: ±16 g), a triaxial gyroscope (full scale: ±2000 dps), and a triaxial magnetometer (full scale: ±50 Gauss). Data was acquired at a sampling frequency equal to 100 Hz (Figure 1). In addition, each participant was equipped with a smartphone (mod. Samsung S9), used as reference system, to determine whether the participant was indoor or outdoor by exploiting the GPS signal acquired at a sampling frequency of 1 Hz. The two systems were synchronised based on their Unix epoch time previously set using a laptop.

A picture containing black, darkness

Description automatically generated

*Figure 1: Experimental setup.*

* 1. GPS-based algorithm (Aeqora App)

The raw GPS coordinates collected during the 2.5-hour observation period need to be aggregated into high-level ground truth labels that characterise the subjects’ activity context. A multi-tier processing framework based on OpenStreetMap [11] (OSM) and a Solr database, is responsible for producing stay-points, that are, locations where subjects spend an extended amount of time without changing their position significantly. Stay-points are detected by either accounting for the absence of GPS data, that usually indicates that the phone does not attempt to acquire location updates as GPS coordinates are deemed stationary or applying distance and time thresholds to groups of proximate consecutive GPS coordinates, hence defining candidate stay-points as centroids of spatial and temporal clusters. Activities performed sufficiently close to a stay-point are labelled as indoor, whilst activities performed away from any stay-point are defined as outdoor. The result of aggregating GPS coordinates is then a set of binary labels, produced with a frequency of 1 Hz, that describe the context of the corresponding IMU signal. Each label represents the continuous probability of that particular activity being performed indoor (value of 1) or outdoor (value of 0).

* 1. Machine learning classification models

Machine learning classification techniques can be divided into supervised and unsupervised depending on the method used during the training phase [4]. Unsupervised learning is used when no labelled data is available, while supervised learning takes labels for various sets of features during the training phase to train the classifier. Given the binary classification nature of this paper, i.e., discerning indoor from outdoor environments, this study employed two distinct methodologies for classification purposes: tree-based models, namely Random Forest (RF) and XGBoost (XGB), and deep learning models, specifically Stacked Long Short-Term Memory (LSTM) networks.

* + 1. Random forest

Random forest is a popular machine learning algorithm that is widely used for both classification and regression tasks [5,6]. It combines multiple decision trees, each of which is trained on a random subset of input data, to make predictions. During the training process, for each node in each decision tree, the algorithm selects a random subset of features to consider when making a split. Then it selects the best feature from the subset and makes a split based on that feature. This process is recursively repeated until a stopping criterion is achieved, such as reaching a maximum tree depth or when there are too few samples left on the leaf node.

Random forest has several advantages with respect to other machine learning algorithms, especially for tabular data: *i)* it can handle both categorical and continuous data, making it suitable for a wide range of datasets; ii) it can handle missing values and outliers, as it only uses a subset of the features at each split; *iii)* it is less likely to overfitting since it combines multiple trees to make a prediction in such a way as to minimise the impact of individual trees that may be overfitting the data; and *iv)* it can provide information on feature importance, which can be useful to understand the underlying relationships in the data and therefore select the most important feature for the prediction.

The above mentioned properties make the random forest a valuable algorithm for discerning indoor from outdoor environments.

* + 1. Extreme gradient boosting

Extreme gradient boosting (XGBoost) is a machine learning algorithm that has gained popularity in recent years for its high performance and scalability. It is a kind of gradient boosting algorithm that uses decision trees as its base model.

The main advantage of XGBoost is its ability to handle large datasets with a high number of features, making it suitable for many real-world scenarios, such as recommender systems, fraud detection, and image classification [7]. Additionally, XGBoost can handle both regression and classification problems, and it has been shown to outperform other popular machine learning algorithms, such as random forest and logistic regression, on certain types of datasets [8,9]. Furthermore, XGBoost has a range of useful features, such as regularisation techniques, missing value handling, and feature importance analysis, that make it a powerful tool for data analysis and prediction.

In this work, the use of an enhanced version of the gradient boosting machines is given by the fact that fairly good results were obtained in [3].

* + 1. *Stacked long short-term memory*

Stacked long short-term memory (LSTM) is a type of recurrent neural network (RNN) architecture that is often used for classification of time series data [10]. Unlike previous algorithms, stacked LSTM is a deep learning algorithm that can model complex temporal dependencies in the data and is capable of capturing long-term patterns in the time series. The stacked LSTM architecture consists of multiple layers of LSTM units, where each layer processes the input sequence and passes its output to the next layer. The output of the final layer is then fed into a fully connected layer for the classification. One advantage of the stacked LSTM over traditional RNNs is that it can learn hierarchical representations of the input sequence, allowing it to model more complex temporal patterns. Additionally, stacked LSTM can effectively handle long-term dependencies in the time series, which is a common challenge in time series classification. To use stacked LSTM for time series classification, the input data is typically pre-processed into a sequence of fixed-length time windows, with each window representing a single observation in the time series. The stacked LSTM model is then trained on this input sequence and corresponding class labels. Figure 2 shows the architecture implemented to manage data recorded by the MIMUs.

A picture containing black, darkness

Description automatically generated

*Figure 2: Stacked LSTM architecture.*

* 1. Data pre-processing

Data acquired by MIMUs (fs=100 Hz) were joined on a per-second basis with the one acquired by the smartphone (fs = 1 Hz) and labelled as indoor or outdoor based on contextual information. Since this work is only focused on the magnetometer data, data acquired by the accelerometer and the gyroscope will not be considered. The result of this inner-joining procedure can be seen in the following Table 1.

*Table 1: result of the inner joining process to have a compact tabular view of the preprocess data. Timestamps range from id ts=0 to ts=N.*

| Timestamps | Feature space | Indoor/Outdoor label |
| --- | --- | --- |
| ts\_0.00 | float value(s) | binary value |
| … | … | … |
| ts\_0.99 | float value(s) | binary value |
| ts\_1.00 | float value(s) | binary value |
| … | … | … |
| ts\_N.99 | float value(s) | binary value |

* + 1. Signal handling

Vector *Si*, as defined in (1), includes triaxial local magnetic field (*B*) data from all MIMUs locations, resulting in 12 features per sample *i*.

(1)

In addition to MIMUs raw signals, the norm has been computed and then added to the existing features set, resulting in a total of 16 features per sample:

(2)

To provide context about the behaviour of the features over time, a dedicated window is applied to the entire dataset. This produces a new dataset that can be viewed as a time series, with a dimensionality of *N*, *F*, *T*, where *N* represents the number of data points, *F* the number of features considered, and *T* corresponds to the window size (1 second and 2 seconds). Specifically, when all MIMUs are taken into account *F* is set to 16, while when considered separately *F* is equal to 4.

It is important to note that the present study does not limit the window selection to only 2 seconds as in [3], but also explored a small window size (i.e., 1 second) to investigate how it affects the performances of the algorithms.

* + 1. Feature engineering

In traditional machine learning algorithms, feature engineering is a crucial step that involves the selection and extraction of the most important features from a dataset. However, in deep learning, feature engineering is not necessary as the model can learn directly more or less hidden feature spaces from the raw input data. In fact, attempting to perform feature engineering on the input data in the latter case may even hinder the performance of the deep learning model. Therefore, when we deal with this feature set, we are referring only to the aforementioned tree-based machine learning algorithms (i.e., random forest, XGBoost).

Due to the characteristics of the local magnetic field phenomenon additional features with respect to those considered by [3] are take into account for a more accurate exploration. The full list of features is provided in Table 2.

*Table 2: List of extracted features.*

| State-of-the-art [3] | Additional |
| --- | --- |
| Mean, median, variance, mean absolute deviation, kurtosis, skewness, percentiles (1, 10, 25, 50, 75, 99), interquartile range, trimmed mean (10%). | Root mean square, sensor correlation, first and second dominant frequency, power at the first and second dominant frequency. |

* + 1. Subject selection

To prevent unbalanced data from affecting classifier knowledge during the training procedure, a proper filtering process is necessary. This process involves the selection of valid participants from the initial dataset. Exclusion criteria are duration of the experimental session less than 75 minutes, gaps in the GPS raw signal, and indoor/outdoor probability equal to 50% (i.e., same probability to be indoor and outdoor). Finally, 20 out of 59 participants have been selected for a total of about 40 hours of data recordings. Therefore, according to the literature (i.e., ratio of 60 to 40 for binary classification to prevent overfitting) [4,7], participants were divided into training set (15 participants) and testing set (5 participants).

* + 1. Dimensionality reduction

Prior to using the features space for training and validating the machine learning models, it's important to consider reducing the dataset's dimensionality. Principal component analysis (PCA) is a useful mathematical function for exploring projection spaces and identifying which features provide the most useful information. Statistical algorithms such as XGB and random forest can benefit from this reduction, as it helps to create better decision trees with less redundant information.

In this work, PCA was not used to process raw local magnetic field signals, as it was not involved in deep learning network analysis, but it was exploited to reduce the high dimensionality brought by the temporal-frequency feature space. The explained variance ratio, which ranges from 0 to 1, is the key value used to select the right number of PCA dimensions. For this study, the heuristic value was set to 0.85, resulting in a total of 12 features.

Figure 3 shows the complete pipeline to achieve the preprocessing step.

A picture containing text, screenshot, diagram, design

Description automatically generated

*Figure 3: Schematic view of the pre-processing pipeline.*

* + 1. Training and testing

The proposed classification approach's performance was evaluated using the leave one subject out cross-validation (LOSO). This process consists in training the models using all subjects except the one to be tested and repeating the process for all subjects in the dataset (Figure 4). This approach helps evaluate the methodology's inter-subject generalization capability. The correct classification accuracy resulting from LOSO cross-validation was used to assess the hyperparameter tuning, which was evaluated using confusion matrices.

The training process for a stacked LSTM involves optimizing a loss function, in this case as binary cross-entropy, using backpropagation through time to update the weights of the LSTM layers and the fully connected layers. This process requires a large amount of training data (the final dataset relies on about 36 hours of data recordings) and can be computationally expensive, especially for large models and datasets. On the other hand, fitting traditional models such as random forest and XGBoost involves using machine learning algorithms to learn patterns and relationships in the input data. These models are typically trained using gradient descent or other optimization algorithms to minimize a loss function. The training process for these models is generally faster and less computationally intensive than training a stacked LSTM, but may not be as effective for certain types of sequence prediction tasks [11]. Hence, for this work it is interesting to explore deep learning models as they possess the capability to discern latent patterns which are indistinguishable through conventional pre-processing techniques.

The methodology was validated offline on a personal computer running Python, with the CUDA environment available for training the stacked LSTM. It was equipped with a 4GB NVIDIA 1650 Ti for faster training inference.

A picture containing text, screenshot, diagram, font

Description automatically generated

*Figure 4: Training, validation, and testing pipeline.*

* + 1. Performance evaluation

The use of accuracy for evaluating binary classification models has several disadvantages that make it not sufficient as the sole metric for performance evaluation. Firstly, accuracy does not take into account the different costs associated with false positive and false negative predictions. In many real-world applications, the cost of a false positive (type I error) may be different from the cost of a false negative (type II error). For instance, in medical diagnosis, a false negative could result in delayed treatment or misdiagnosis, leading to adverse health outcomes, while a false positive could result in unnecessary tests or treatments. Therefore, accuracy alone cannot provide a complete picture of the model's performance in such scenarios. Secondly, accuracy is not appropriate when the data is not balanced, i.e., when one class is represented more than the other. In such cases, a model that always predicts the majority class will achieve high accuracy, but this may not be desirable. Additional metrics, such as F1-score or AUC ROC that consider both classes equally, should be considered. Thirdly, accuracy does not account for the uncertainty in the model's predictions. In some cases, a model may have high accuracy but low confidence in its predictions, making it less reliable than a model with lower accuracy but higher confidence.

In summary, relying solely on accuracy for evaluating binary classification models can be misleading and may not provide a complete picture of the model's performance. Therefore, we will exploit other metrics, such as F1-score and AUC ROC, in order to complement accuracy and provide a more comprehensive evaluation of the models.

F1-score is a measure of the trade-off between precision and recall, and it takes into account both the false positive and false negative rates of the model. AUC ROC, on the other hand, is a performance metric that measures the model's ability to correctly classify positive and negative samples. It provides a single score that summarizes the model's ability to distinguish between the two classes across all possible classification thresholds.

1. RESULTS
2. Models’ performances
3. Ablation studies
4. DISCUSSION
5. CONCLUSIONS
6. SUPPLEMENTARY MATERIALS
7. AUTHOR CONTRIBUTIONS
8. FUNDING
9. DATA AVAILABILITY STATEMENT
10. ACKNOWLEDGMENTS
11. CONFLICTS OF INTEREST
12. REFERENCES

[1] K. Scott, L. Alcock, E. Buckley, C. Hansen, C. Becker, P. Brown, Design and validation of a multi-task , multi-phase protocol for real-world gait simulation, (n.d.).

[2] F. Salis, S. Bertuletti, K. Scott, M. Caruso, T. Bonci, E. Buckley, U. Della Croce, C. Mazzà, A. Cereatti, A wearable multi-sensor system for real world gait analysis, (n.d.).

[3] I. Ashraf, S. Hur, Y. Park, MagIO: Magnetic field strength based indoor- outdoor detection with a commercial smartphone, Micromachines. 9 (2018). <https://doi.org/10.3390/mi9100534>.

[4] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

[5] Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32. [6] Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. R News, 2(3), 18-22. [7] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16), 785-794. [8] Li, Z., & Lin, Y. (2020). Comparison of machine learning algorithms for real-time predictive modelling of flood events. Journal of Hydrology, 586, 124858. [9] Zhang, Y., Sun, G., & Luo, Y. (2021). Comparative study of machine learning algorithms for predicting road traffic crashes. Accident Analysis & Prevention, 155, 106150. [10] Wang, Y., Yan, Y., & Oates, T. (2017). Time series classification from scratch with deep neural networks: A strong baseline. In Proceedings of the 2017 SIAM International Conference on Data Mining (SDM '17), 561-569. [11] Bagnall, A., Lines, J., Bostrom, A., Large, J., & Keogh, E. (2017). The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. Data Mining and Knowledge Discovery, 31(3), 606-660.

1. APPENDIX