Tiako P.F. (ed) *Smart and Sustainable Applications*. Chronicle of Computing. OkIP

https://doi.org/10.55432/978-1-6692-0005-5_9

Machine Learning-based Positioning using Multivariate Time Series Classification for Factory Environments

Nisal Hemadasa[a], Marcus Venzke[a], Volker Turau[a], Yanqiu Huang[b]

[a]*Hamburg University of Technology, 21073 Hamburg, Germany*
[b]*University of Twente, 7522 NB Enschede, Netherlands*

ABSTRACT

Indoor Positioning Systems have gained significance in numerous industrial applications. While state-of-the-art solutions are accurate, their reliance on external infrastructures can lead to considerable costs, deployment complexities, and privacy concerns, making them suboptimal for specific contexts. Recent advancements in machine learning have surfaced as a potential solution, leveraging data solely from onboard IoT sensors. Nonetheless, the optimal machine learning models for IoT's resource constraints remain uncertain. This research introduces an indoor positioning system using motion and ambient sensors tailored for factories and similar settings with predetermined paths. The problem is framed as multivariate time series classification, comparing various ML models. A novel dataset simulating factory assembly lines is utilized for evaluation. Results demonstrate models achieving over 80% accuracy, with 1 Dimensional-Convolutional Neural Networks showing the most balanced performance followed by Multilayer Perceptrons, considering accuracy, memory footprint and latency. Decision Trees exhibit the lowest memory footprint and latency, rendering its potential for practical implementation.

Keywords: Indoor Positioning, Machine Learning, Sensor Fusion, Multivariate Time Series Classification

I. INTRODUCTION

Indoor Positioning System (IPS) is a technology widely adopted in many industries (Farahsari et al., 2022). It is among the foremost in technological fronts such as Smart Cities, Industrial Internet of Things (Frank, 2022). In these areas, IPSs are vital in tracking, navigation, proximity, and inertial measurements, enhancing efficiency, accuracy, and safety in processes.

Higher animals naturally navigate and locate themselves using cues like geomagnetic fields, celestial bodies, wind direction, temperature, scent, and visual landmarks. They form mental paths through learning and memory, aiding in path finding, recognizing environments, and distinguishing locations. This concept can be applied to localizing entities along a predetermined path. Processing sensory inputs allows estimating the current position over time. This can be perceived as a sub-problem of Indoor Positioning. However, unlike the conventional IPSs, where precise x-y coordinates are estimated, we reframe the problem to determine a relative segment on a predefined path.

Numerous research works address the indoor positioning problem from various approaches. These approaches provide precise x-y coordinates of estimated locations. Many of these methods rely on an external infrastructure for reliability, such as (Received Signal Strength Indication) RSSI-based solutions needing consistent signal coverage. However, for applications involving relative positions, like tracking goods in an assembly line, x-y coordinates are less significant. Further, supplementary infrastructure deployment is costly and technically challenging, such as in tunnels or mining sites. Vision-based methods without infrastructure raise privacy concerns. Some methods rely on external information, causing accumulated errors like dead reckoning's starting point. Furthermore, some methods assume Gaussian noise or linear motion dynamics.

To address these limitations, our research explores learning indoor positions from sensor data for Machine learning (ML) model execution on low-power devices like microcontrollers. We combine inertial sensors (accelerometer, gyroscope, and magnetometer) with ambient sensors (pressure, temperature, humidity, and spectrum) for this purpose. These sensors' recordings form multivariate time series data, fused to derive accurate location estimates along a predefined path. The indoor positioning task is therefore formulated as a Multivariate Time Series Classification (MTSC) problem. ML is used to extract sensor data information without making any assumptions about noise or motion dynamics, ensuring accurate functionality without prior information. Challenges arise from resource-constraint hardware and maintaining precision in changing environmental conditions.

The problem generally addressed above could be more specifically described using an assembly line in a factory. Assembly lines consist of preset routes for time and cost optimization. Movement of goods along these routes demands real-time tracking. Modern production lines incorporate digitization and enterprise resource planning (ERP) systems. Especially the process automation in industry 4.0 necessitates asset localization in secure factory environments adhering to strict privacy protocols.

Presented at the 2023 OkIP International Conference on Automated and Intelligent Systems (CAIS) in Oklahoma City, OK, USA, and Online, on October 2-5, 2023

To this end, we evaluate lightweight ML models such as Decision Trees (DT), random forests (RF), for constrained devices such as low-end Microcontroller Units (MCUs) with under 10kB SRAMs (Kumar et al., 2017). Furthermore, we explore complex Long Short-Term Memory networks (LSTM) for indoor positioning via time series classification (Yu et al., 2021). Additionally, we employ state-of-the-art time series classification benchmark architectures as baselines namely, Multilayer Perceptrons (MLP), Convolutional Neural Networks (CNN) (Fawaz et al., 2019), to compare the performance with the prior mentioned models.

To the best of our knowledge, this work pioneers formulating indoor positioning as an MTSC problem, fusing motion and ambient sensors without x-y coordinates for settings like factories. It also investigates ML model feasibility given hardware constraints to estimate relative position. Furthermore, state-of-the-art Deep Neural Network (DNN) models for Time Series Classification (TSC) (Wang et al.,_2017; Fawaz et al., 2019) serve as baselines for the formulated models, using a new indoor positioning time series dataset.

The contributions of this work are as follows.

- Utilization of motion and ambient sensor measurements for entity localization on a known path, encompassing indoor and outdoor segments, without requiring external infrastructure. The significance and limitations of each sensor type are examined.
- Introducing a novel multivariate time series dataset comprising sensor data from IMU, pressure, temperature, humidity, and spectrum sensors, collected during traversal of three distinct paths.
- The localization on a known path is treated as an MTSC problem. The performances ML models, namely DT, RF, MLP, CNN, LSTM in solving it, are compared against the baseline models, mainly based on accuracy, memory footprint, inference latency.

The rest of this paper is organised as follows. Section II presents the related works. Section III generalizes the indoor positioning task as an MTSC problem. Section IV introduces the ML models evaluated in this work. Section V describes the novel multivariate time series dataset and a case study. Section VI presents model evaluations and analysis. Finally, Section VII concludes the work.

II. Related Work

The existing work on indoor positioning are commonly classified technology-wise, technique-wise and algorithm-wise (Hayward et al., 2022; Yang et al., 2021; Wu et al., 2018; Ouyang & Abed-Meraim, 2022; Sesyuk et al., 2022; Poulose et al., 2019; Pascacio et al., 2021). Technology-wise solutions encompass satellite-based, radio communication-based, visible light-based, inertial navigation-based, magnetic-based, sound-based, and vision-based methods. However, many solutions in this range lack privacy and independence from external infrastructure like wireless access points (APs).

While privacy-centric wireless technology-based positioning systems exist (Holcer et al., 2020), they often require supplementary infrastructure. For certain applications like tracking assembly line goods, relative location matters more than precise x-y coordinates. However, these solutions primarily focus on determining x-y coordinates.

Collaborative use of different technologies also exists (Pascacio et al., 2021). Magnetic field-based localization solutions are infrastructure-independent and privacy-secure, offering stability and unique magnetic signals (Chiang et al., 2020). However, they are sensitive to electromagnetic disturbances like motion of metallic structures near magnetometers. Hence, the magnetic-field based solutions are often fused with other sensors.

Work classified under technique-wise includes dead-reckoning, vision analysis, triangulation, fingerprinting and proximity-based. The latter three techniques require external infrastructure for their operation. Dead-reckoning requires external data such as initial coordinates and accumulates error. Vision based systems can compromise privacy.

Algorithm-wise categorizes into least square, maximum likelihood, deterministic/probabilistic methods (Pascacio et al., 2021). Fusion-based methods vary from conventional (least squares, maximum likelihood, etc.) to state estimation (hidden Markov, Kalman, etc.) and ML (k-nearest neighbors, RF, SVM, NN) approaches (Guo et al., 2020). Algorithm-wise classification mainly branches to the least square method, maximum likelihood method, deterministic or probabilistic method (Pascacio et al., 2021). The existing fusion-based methods range from conventional (least squares, maximum likelihood, maximum a posterior, and minimum mean squares error), to state estimate (hidden Markov model, Kalman filter, extended Kalman filter, and particle filter), and ML methods (k-nearest neighbours, RFs), support vector machine, and neural networks) (Guo et al., 2020). ML has been applied mainly to RSSI and fingerprint-based systems (Guo et al., 2020; Nessa et al., 2020). They often assume Gaussian noise and linear motion (Nessa et al., 2020).

III. Problem Definition

Localization of a moving or stationary asset in production lines, factories and warehouses, at a given time is inferred by identifying the probable segment of a known path. This can be interpreted as a classification problem. As assets traverse the path, attached motion and ambient sensors gather periodic measurements, forming multivariate time series. These data are fed into the edge device for real-time classification. In general terms, the problem can be defined as follows.

Definition 1. *Let P be a path partitioned into l segments, such that* $P \rightarrow [\mathrm{s}_1, \dots, \mathrm{s}_\mathrm{m}, \dots, \mathrm{s}_\mathrm{l}]^T$. *For each* $m \in \{1, \dots, l\}$ *segment* s_m *is uniquely identified by a label* y_m *from a set* $Y_P = \{y_1, y_2, \dots, y_m, \dots, y_l\}$ *of labels of P.*

An asset moves along path P. The speed and its variations remain unknown. Further, the time it takes to complete the path differ across instances. Thus, no trivial position-time correlation exists. Path P has segments for both forward and return journeys, resembling a round trip. However, it excludes anomalies like extended pauses or deviations from the path, which are common in industrial environments.

Definition 2. *Let X^i denote a univariate time series of a feature i, engineered from the recordings of sensor measurements, as a result of traversing a complete path P. An observation at a given sampling time t is denoted as x_t^i and $X^i = [x_1^i, x_2^i, \dots, x_k^i]^T$, where k is the total number of observations of time series X^i.*

Definition 3. *Let there be n different features, giving distinct univariate time series for a data-collection run along P. Then $X = [X^1, X^2, \dots, X^n]$ is then defined as a multivariate timeseries for P.*

Finally, we formally define the problem addressed in this work.

Definition 4. *Let $X_t = [x_t^1, x_t^2, \dots, x_t^n]$ be the observations of all features in the feature space, at a given time t. The problem addressed in this work is to find a function $f_P^j: \{X_t, X_{t-1}, \dots, X_{t-j}\} \to Y_P$, that determines the label y_m of the segment in which the object is at time t, using time series $[X_t, X_{t-1}, \dots, X_{t-j}]$, where j is a predetermined window size, such that $j \epsilon \mathbb{Z}_0^+$, and for $j \leq t-1$ it holds that $f_P^j(X_t, X_{t-1}, \dots, X_{t-j}) = y_m$.*

IV. IPS USING DT, RF, CNN AND LSTM NETWORKS

In this section, we outline the architectures of the ML models employed to learn the function f_P^j as defined in Definition 4. We use two recently popular time series classification baseline models namely, MLP, and Fully Convolutional Networks (FCN) (Fawaz et al., 2019; Wang_ et al., 2017). Additionally, a tree-based approach, DT with entropy is applied, known for its relative simplicity and suitability for low-performance edge devices like MCUs. An ensemble approach, RF is also explored. Lastly, we assess the dataset's performance using vanilla LSTM, bidirectional LSTM (BiLSTM), CNN-1D, and CNN-2D for solving the TSC task. For the latter two, we add dense layers to further enhance the performance rather than developing it to fully convolutional architectures. Detailed model architecture descriptions are available in (Hemadasa et al. 2023).

The runtime environment for many related indoor positioning works is resource-constrained (Nessa et al., 2020). Thus, optimizing ML models requires considering both accuracy and resource usage. Therefore, in this study, we assess simpler, lighter models that maintain notable accuracy, aligning with the problem defined in Section III.

V. DATASET DESCRIPTION: MOTION-AMBIENT DATASET

In this section, we present a use case scenario to validate our claims using the novel *Motion-Ambient* dataset. We simulate a practical factory and warehousing scenario within Hamburg University of Technology premises. A portable data-logging setup gathers IMU, pressure, humidity, temperature, and spectrum data along three diverse paths, encompassing indoor and outdoor segments. These paths feature various dynamics like indoor passages, elevators, ramps, stairs, changing terrain roughness (e.g., cobblestone), different lighting conditions, and magnetic interference near metal structures, which are often encountered in industrial environments. The paths are simultaneously annotated with predetermined segments (classes) as the setup is transported.

Motion-Ambient is a time series dataset designed for benchmarking indoor localization research. Detail description of this dataset and its preprocessing is available in (Hemadasa et al. 2023).

VI. RESULTS, ANALYSIS AND DISCUSSION

In this section, we describe constraints for structuring ML models' architectures. Additionally, we introduce the metrics used to compare the results and to derive further insights.

A. *Description of model architectures and their constraints*

Apart from its use in regression problems, decision trees classify data by features, while dense layer networks such as MLP do the same and additionally identifying complex, non-linear patterns. RF is employed for similar purposes, typically offering higher accuracy and unbiased predictions compared to decision trees, but demanding more computations and memory usage. Initially designed to recognize spatial patterns in images, CNNs can also be applied to MTS data organized as a 2D heat map, enabling them to extract local temporal patterns across time and features. The combination of CNNs with dense layers simplifies the learned patterns into a classification task. Moreover, CNNs generate large feature spaces where dense layers learn complex patterns and correlations among these features. LSTM models, in contrast, capture long-term dependencies between time steps in time series data. This differs from CNNs, which identify patterns within local regions of a multivariate time series, while LSTMs learn relationships among multiple of these regions. Unlike LSTMs, DTs, RFs, and MLPs aren't primarily tailored for capturing time correlations in data. However, utilizing a time window of data does enhance accuracy, indicating their limited capability in detecting patterns across time. We consider MLP and FCN models proposed by Wang et al. (2017) to benchmark the neural network models that we present.

The timestep window size of j=30 yields the highest accuracy out of {10, 20, 30, 40, 50}, for benchmarking models MLP and FCN. Above this value, the models tend to overfit, and below it, underfitting is observed. This window size is employed consistently across all models. Detailed architectural justifications are elaborated in (Hemadasa et al. 2023).

B. *Accuracy Metrics*

In this work, along with the ML accuracy-score, we utilize a specialized accuracy metric named Loc-score.

1. *Accuracy-score*

ML classification accuracy-score (accuracy) is the ratio of correct predictions to total predictions, as shown in equation (1). This work equates it to the proportion of sensor samples accurately classified against their assigned annotations.

$$Acc.score = \frac{Total\ number\ of\ correct\ predictions}{Total\ number\ of\ predictions} \quad (1)$$

However, the labels used here might not entirely align with the ground truth due to labeling noise. This is primarily due to the sensors' high sampling rates, surpassing human reaction times

TABLE I. ACCURACY-SCORES ACROSS THE ML MODELS. THE HIGHEST ACCURACY CORRESPONDING TO EACH PATH IS HIGHLIGHTED.

Path	MLP	FCN	DT	RF	LSTM	BiLSTM	CNN-1D	CNN-2D
1	0.8848	0.8301	0.8394	0.8977	0.8135	0.8445	**0.9105**	0.8635
2	0.952	0.9298	0.8405	0.8946	0.8735	0.8872	**0.9544**	0.8891
3	**0.9321**	0.9185	0.8777	0.9217	0.9116	0.9013	0.9302	0.8939

TABLE II. LOC-SCORES ACROSS THE ML MODELS. THE HIGHEST LOC-SCORE CORRESPONDING TO EACH PATH IS HIGHLIGHTED.

Path	MLP	FCN	DT	RF	LSTM	BiLSTM	CNN-1D	CNN-2D
1	0.9079	0.8542	0.8529	0.9199	0.827	0.8205	**0.9315**	0.8367
2	0.966	0.9465	0.845	0.9108	0.8932	0.9058	**0.9679**	0.9058
3	**0.9438**	0.9312	0.9033	0.935	0.9244	0.9164	0.9416	0.9057

TABLE III. MEMORY FOOTPRINT OF THE ML MODELS IN MB. THE BEST MEMORY FOOTPRINT CORRESPONDING TO EACH PATH IS HIGHLIGHTED..

Path	MLP	FCN	DT	RF	LSTM	BiLSTM	CNN-1D	CNN-2D
1	7.72	18.72	**0.99**	24.65	30.92	75.96	5.37	67
2	7.63	18.72	**0.44**	11	30.87	75.74	5.36	66.5
3	7.65	9.93	**0.72**	17.98	30.88	75.78	5.35	25.65

and making consistent annotation of path segment transitions challenging across data-collection runs.

To address labeling inconsistencies, Grewe introduces the accuracy metric loc-score (Grewe, 2021).

2. Loc-score

Loc-score defines a window of timesteps around transitions from one segment (considered as a class in this problem) to the next, in the true class labels. During the evaluation, predictions to either of the two classes, within this window, are considered correct, while prediction to other classes are considered misclassified. The ratio of the samples consequently correctly predicted and the total number of predictions is defined as the Loc-score. This can be more formally defined as follows.

Definition 5. *For a transition from a segment* y_m to y_{m+1}, *at a given timestep* t_{tr}, *and a defined window size of* $2\tau + t_{tr}$, *a classification* $\widehat{y_t}$, *at time* t, *is considered correct only if* $\widehat{y_t} \in \{y_m, y_{m+1}\}$, *such that* $t \in [t_{tr} - \tau, t_{tr} + \tau]$. *Then,*

$$Locscore = \frac{Number\ of\ correct\ predictions\ (per\ Def.5)}{Total\ number\ of\ predictions} \quad (2)$$

3. Memory Footprint

The memory footprint of a trained ML model is the memory needed to store the network's parameters including structure, trained weights, and biases of all layers.

4. Inference Latency

Inference latency is the time taken for a ML model to make a prediction based on input data.

5. Throughput

Throughput is the prediction rate, i.e., predictions per ms.

C. Analysis of Results

1. Accuracy

Table I and Table II show that the CNN-1D model achieves the highest accuracy and loc-score for both path 1 and 2. Path 1 sees CNN-1D closely followed by RF and MLP models, respectively. MLP excels in accuracy for path 3, slightly ahead of CNN-1D, while RF lags behind. On path 2, RF's accuracy is significantly lower than MLP and CNN-1D, even trailing FCN. Following are FCN, BiLSTM, LSTM, and CNN-2D, with rankings varying per path. DT maintains the lowest average accuracy, except for path 1.

For j=1, both DT and RF had notably lower accuracy-scores (0.7737, 0.7873, 0.8264 and 0.8216, 0.8722, 0.8583 respectively) and loc-scores (0.7959, 0.7981, 0.8422 and 0.8483, 0.8901, 0.8732 respectively) for paths 1, 2 and 3 compared to j=30. This suggests that tree-based models can partially capture time correlations despite their design limitations. Enhancing feature engineering could further improve the accuracy of DT and RF.

In summary, architectures like MLP, CNN-1D, and RF that capture short-to-mid range time dependencies achieve higher accuracy. The limited impact of the two LSTM variants on classification accuracy suggests that the dataset might lack significant long-term patterns they excel at identifying. For paths 2 and 3, DT's temporal correlation extraction falls short compared to other models in capturing feature dynamics.

However, the dataset for path 1 has less complex, more structured data making it easier for both DT and RF to capture. This is evident from the relatively higher accuracy values of DT and RF for path 1 compared to the other paths.

TABLE IV. INFERENCE LATENCY (MS) OF DIFFERENT ML MODELS WITH STANDARD DEVIATION. LOWEST INFERENCE LATENCY FOR EACH PATH IS HIGHLIGHTED.

Path	MLP	FCN	DT	RF	LSTM	BiLSTM	CNN-1D	CNN-2D
1	0.053±0.003	0.359±0.012	**0.002±0**	0.012±0.003	0.375±0.009	0.413±0.028	0.05±0.005	0.263±0.008
2	0.058±0.009	0.387±0.012	**0.002±0**	0.007±0.001	0.382±0.017	0.41±0.03	0.05±0.001	0.26±0.011
3	0.052±0.01	0.208±0.016	**0.001±0**	0.007±0	0.368±0.009	0.376±0.007	0.05±0.005	0.121±0.013

TABLE V. INFERENCE THROUGHPUT OF ML MODELS IN PREDICTIONS PER MS, WITH STANDARD DEVIATION. HIGHEST THROUGHPUT VALUES FOR EACH PATH ARE HIGHLIGHTED.

Path	MLP	FCN	DT	RF	LSTM	BiLSTM	CNN-1D	CNN-2D
1	19.03± 1.37	2.79± 0.09	**569±77**	91.09± 24.69	2.67± 0.07	2.43± 0.15	20.29±2.19	3.81±0.11
2	17.46±2.24	2.58±0.08	**644±35**	136.4±9.27	2.62±0.11	2.45±0.16	19.92±0.46	3.85±0.16
3	19.98±3.24	4.83±0.37	**1080±114**	147.5±8.49	2.72±0.06	2.66±0.05	20.24±1.59	8.37±0.92

CNN-1D achieves higher accuracy than CNN-2D across all paths, possibly due to architectural enhancements like batch normalization, which aids model regularization and accuracy in CNN-1D. Moreover, Conv1D's feature extraction might be more effective compared to Conv2D. Conv1D's convolution along sequential feature vectors could derive more informative features than simultaneous convolution across parts of the feature space and their temporal dynamics in Conv2D.

Despite LSTM and BiLSTM being architectures capable of extracting complex temporal correlations, they do not produce the best results. To understand why, initially, the hyperparameters are varied within the limits of a maximum of 3 layers with 64 cells in each layer and j=30. This does not improve accuracy. In conclusion, we are left with several reasons such as the dataset having simple temporal complexities, the amount of data required for the models to learn being simply too small, the time step window size of j=30 being not large enough, the lesser number of layers results in a too shallow network in the case of BiLSTM, etc., which we do not cover in this paper.

Despite LSTM and BiLSTM's potential for extracting complex temporal correlations, the reason for them not yielding optimal results could be due to factors like the dataset's simplicity in terms of temporal complexities, inadequate data for effective learning, the relatively smaller time step window size of j=30, and the shallowness of BiLSTM networks due to fewer layers might contribute to these results.

2. Memory Footprint

Table III shows the notably minimal memory footprint of DT across all three paths, significantly surpassing the others. Conversely, the BiLSTM model exhibits the highest memory usage. Notably, for path 3, the FCN and CNN-2D models have reduced memory footprints compared to paths 1 and 2. This difference can be attributed to the smaller input feature space of path 3 (9), in contrast to the other paths (17), as FCN and CNN-2D models depend on input feature space size. More details about the features are available in (Hemadasa et al. 2023).

DT models' memory footprint can be accommodated by high-end MCUs like ESP32, which has 500kB SRAM and 4MB flash memory on average, even without additional compression methods such as swapping (Miao & Lin, 2021). Pruning further reduces their size, facilitating deployment in highly resource-constrained MCUs without significant accuracy loss (Kumar et al., 2017). In edge ML, the device's size impacts more than the model architecture, on the energy budget (Banbury et al., 2021a). This can reduce energy consumption fitting them well for long-term industrial positioning applications. Similarly, other DNN models can be adapted for high-end MCUs by optimizing for constrained hardware (Banbury et al., 2021b; Fedorov et al., 2020), though with potential accuracy trade-offs.

3. Inference Latency and Throughput

In this study, inference latency and throughput are evaluated on an Intel Core i5-10210U CPU with a base clock of 1.60 GHz, which can boost up to 2.10 GHz under load. This system has 16 GB memory and runs Windows 10. These specifications represent the higher end of current edge devices' performance spectrum. Prior to each inference run, the device is rebooted and only the inferencing application operates in JupyterLab. The inference dataset size remains constant at 100,000 inputs for all models, with each input consisting of 30 timesteps. The average time for this set of predictions is computed to establish the time per prediction in milliseconds (ms). Additionally, each model undergoes 10 tests, and the mean and standard deviation are calculated.

Tables IV and V show that DT is the fastest model across all three paths, significantly outperforming the others. RF comes next in terms of speed. CNN-1D and MLP models follow, closely trailing each other. Subsequently, CNN-2D model ranks higher, followed by FCN and LSTM models with similar latency performances. BiLSTM exhibits the slowest speed compared to the other models.

The IMU sensor, with a data sampling rate of 0.041s (~24Hz), is the fastest among the sensors in the scenario. All assessed models can meet this frequency requirement. This is based solely on inference latency, without considering additional processing overheads. For the slowest processing, MLP, FCN, DT, RF, LSTM, BiLSTM, CNN-1D, and CNN-2D can handle sensor sampling rates of up to 14kHz, 2kHz, 5MHz, 172kHz, 2kHz, 2kHz, 18kHz, and 3kHz, respectively. Given the application's context, all these inference rates surpass the fastest sensor rate. Therefore, considering both inference latency and throughput, all models can be effectively deployed in the described use case scenario using the hardware in these experiments.

Besides model architectures, inference latency and throughput significantly rely on hardware characteristics like processor speed, memory, operating system, possible accelerator hardware, and optimization for deep learning.

D. Discussion and Future Improvements

Data drift, a common challenge in ML methods, arises from gradual sensor changes or shifts in the environment, like modifications in assembly line configurations. Such instances necessitate model retraining with fresh data, marking a drawback compared to many existing IPSs. This becomes limitation for scalability of the proposed approach.

This work focuses on indoor positioning for assets moving along specific paths with variable speeds, completion times, and round-trip routes. To refine this, we aim to enhance models' accuracy amid environmental anomalies. This involves incorporating diverse scenarios like deviations from paths, pauses, collisions, and unexpected events, aiming to enhance algorithm adaptability, exposing the models to unseen data.

We plan to enhance the generalizability of the ML models to classify common indoor motions like ramps, elevators, and turns, broadening their usability with transfer learning, smaller datasets, faster training, and better accuracy. MLP and CNN-1D, performing consistently well, are already suitable for this application.

Temperature, humidity, and spectral attributes could have seasonal effects that could impact classification outcomes. However, our dataset lacks this seasonal representation. Hence, our current study doesn't address accuracy variations across seasons, which will be explored in future work.

We further intend to assess model performance by deploying them on low power and low performance edge devices for real-time on-site data sensing.

VII. Conclusion

This paper explores ML with motion and ambient sensors for indoor positioning in factories and similar contexts. We introduce the new Motion-Ambient dataset containing multivariate time series data. Using this dataset, we frame indoor positioning as an MTSC problem and assess ML models like DT, RF, LSTM, BiLSTM, CNN-1D, and CNN-2D. These models are compared to benchmark algorithms MLP and FCN, evaluated via accuracy, loc-score, memory footprint, inference latency, and throughput. Results indicate accuracy levels exceeding 80%, fitting use cases. All models meet latency demands. The memory footprint spans 0.5 - 76 MB, with CNN-1D and MLP performing optimally. Notably, DT and RF's memory and latency advantages could be further enhanced with manual feature engineering. This study demonstrates ML's pplicability to indoor positioning and plans real-world deployment in a factory environment.

References

Farahsari, P. S., Farahzadi, A., Rezazadeh, J., & Bagheri, A. (2022), A survey on indoor positioning systems for iot-based applications, *IEEE Internet of Things Journal*, vol. 9, no. 10.

Frank´o, A., Holl´osi, G., Ficzere, D., & Varga, P. (2022), Applied machine learning for IIoT and smart production—methods to improve production quality, safety and sustainability, *Sensors*, vol. 22, no. 23, p. 9148.

Kumar, S. Goyal, & Varma, M. (2017), Resource-efficient machine learning in 2 kb ram for the internet of things, *in Proceedings of the 34th International Conference on Machine Learning.*

Fawaz, H. I., Forestier, G., Weber, J., Idoumghar, L., & Muller, P.-A. (2019), Deep learning for time series classification: a review, *Data Mining and Knowledge Discovery*, vol. 33, no. 4.

Hemadasa, N., Venzke, M., Turau, V., & Huang, Y. (2023) Machine Learning-based Positioning using Multivariate Time Series Classification for Factory Environments, *arXiv 2023*, arXiv: http://arxiv.org/abs/2308.11670

Yu, W., Kim, I. Y., & Mechefske C. (2021), Analysis of different RNN autoencoder variants for time series classification and machine prognostics, *Mechanical Systems and Signal Processing*.

Wang, Z., Yan, W., & Oates T. (2016), Time series classification from scratch with deep neural networks: A strong baseline.

Hayward, S., van Lopik, K., Hinde, C., & West A. (2022), A survey of indoor location technologies, techniques and applications in industry, *Internet of Things*, vol. 20, p. 100608, Nov.

Yang, T., Cabani, A., & Chafouk, H. (2021), A survey of recent indoor localization scenarios and methodologies, *Sensors*.

Wu, Y., Zhu, H.-B., Du, Q.-X., & Tang, S.-M. (2018), A survey of the research status of pedestrian dead reckoning systems based on inertial sensors, *International Journal of Automation and Computing*.

Ouyang, G., & Abed-Meraim, K. (2022), A survey of magnetic-field-based indoor localization, *Electronics*, vol. 11, no. 6, p. 864.

Sesyuk, A., Ioannou, S., & Raspopoulos, M. (2022), A survey of 3d indoor localization systems and technologies, *Sensors.*

Poulose, A., Kim, J., & Han, D. S. (2019), A sensor fusion framework for indoor localization using smartphone sensors and wi-fi RSSI measurements, *Applied Sciences*, vol. 9, no. 20, p. 4379.

Pascacio, P., Casteleyn, S., Torres-Sospedra, J., Lohan, E. S., & Nurmi, J. (2021), Collaborative indoor positioning systems: A systematic review, *Sensors*, vol. 21, no. 3, p. 1002.

Holcer, S., Torres-Sospedra, J., Gould, M., & Remolar, I. (2020), Privacy in indoor positioning systems: A systematic review, *International Conference on Localization and GNSS*.

Chiang, T.-H., Sun, Z.-H., Shiu, H.-R., Lin, K. C.-J., & Tseng, Y.-C. (2020), Magnetic field-based localization in factories using neural network with robotic sampling, *IEEE Sensors Journal*.

Guo, X., Ansari, N., Hu, F., Shao, Y., Elikplim, N. R., & Li, L. (2020), A survey on fusion-based indoor positioning, IEEE *Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 566–594.

Nessa, A., Adhikari, B., Hussain, F., & Fernando, X. N. (2020), A survey of machine learning for indoor positioning, *IEEE Access*.

Grewe, T. (2021), Smartphone sensor-based location awareness using decision trees, *Master's thesis, Institute of Telematics, Hamburg University of Technology.*

Miao, H., & Lin, F. X. (2021), Enabling large neural networks on tiny microcontrollers with swapping.

Banbury, C., Zhou, C., Fedorov, I., Matas, R., Thakker, U., Gope, D., Reddi, V. J., Mattina, M., & Whatmough P. (2021a), Micronets: Neural network architectures for deploying tinyml applications on commodity microcontrollers, *Proceedings of Machine Learning and Systems*.

Banbury, C., Reddi, V. J., Torelli, P., Holleman, J., Jeffries, N., Kiraly, C., Montino, P., Kanter, D., Ahmed, S., & Pau et al. D. (2021b), Mlperf tiny benchmark.

Fedorov, I., Stamenovic, M., Jensen, C., Yang, L.-C., Mandell, A., Gan, Y., Mattina, M., & Whatmough P. N. (2020), Tinylstms: Efficient neural speech enhancement for hearing aids, *Interspeech.*