

Received 28 February 2026, accepted 19 March 2026, date of publication 25 March 2026, date of current version 30 March 2026.

Digital Object Identifier 10.1109/ACCESS.2026.3677599

RESEARCH ARTICLE

Predictive Human Trajectory Classification in Industrial Settings With Transfer Learning and Data Augmentation

EVEN FALKENBERG LANGÅS¹, (Graduate Student Member, IEEE), ATLE AALERUD²,
SVEIN OLAV NYBERG¹, MUHAMMAD HAMZA ZAFAR¹, (Graduate Student Member, IEEE),
DANIEL HAGEN¹, (Member, IEEE), AND FILIPPO SANFILIPPO¹, (Senior Member, IEEE)

¹Department of Engineering Sciences, Faculty of Engineering and Science, University of Agder, 4879 Grimstad, Norway

²DARWIN Research Group, NORCE Norwegian Research Center, 4879 Grimstad, Norway

Corresponding author: Even Falkenberg Langås (even.falkenberg.langas@uia.no)

This work was supported by the University of Agder, and by the European Commission Project COGNitive Industries for smart MANufacturing (COGNIMAN) under Grant 101058477.

ABSTRACT The successful realisation of Industry 5.0 relies on seamless collaboration between humans and robots in shared workspaces. This paper investigates predictive human trajectory classification (HTC) in a factory setting to enhance safety, productivity, and adaptability in human-machine environments. Using a state-of-the-art light detection and ranging (LiDAR) system with reflective mirrors to extend the field of view, motion data are collected across a 26 m × 20 m indoor industrial floor. Human trajectories are classified to predict workstation transitions using a stacked long-short term memory (LSTM) network. To address the challenges of limited real-world data, transfer learning is applied by first training the model on synthetically generated trajectories. The paper compares a traditional stochastic method, specifically the Ornstein-Uhlenbeck (OU) process, with a modern learning based method, represented by a timeseries generative adversarial network (TimeGAN). Further on, the algorithm is fine-tuned on the real-world dataset. The comparison shows that in data-scarce scenarios, OU outperforms TimeGAN in most metrics. The LSTM model trained on OU-based data and real-world data shows an overall accuracy gain of 4.8 percentage points compared to training on real-world data only. It is able to classify the goal workstation three seconds in advance with an accuracy of 95.6%, demonstrating the potential for predictive models to improve workplace operations by reducing stall time, optimising workstation readiness, and reducing the need for traditional safety cages.

INDEX TERMS Data augmentation, human trajectory prediction, human trajectory classification, industry 5.0, LiDAR, transfer learning.

I. INTRODUCTION

When people interact, they unconsciously make predictions about each other to foresee scenarios so that they can act appropriately. As our brain develops, this intrinsic ability improves by observing the people around us. Through experience, we learn to understand and predict intentions, thus we can efficiently react to actions by preparing for what will happen.

The associate editor coordinating the review of this manuscript and approving it for publication was Tao Huang¹.

With the advent of Industry 5.0, we see a growing trend in humans and machines sharing workspace, such as in human-robot collaboration [1], and even sharing forces and decisions, such as in human-robot teaming (HRT) [2], [3]. In contrast to humans, machines do not possess any intrinsic sensory or cognitive abilities. Sensors can reproduce human-like visual, aural and haptic sensations. However, human-like cognitive processing of these sensations is a challenge. Seamless collaboration relies on a mutual understanding of intentions, expectations, and the ability to adapt to each other's actions. Therefore, it is crucial to develop algorithms

that incorporate predictive planning and model predictive control to understand and predict human behaviour.

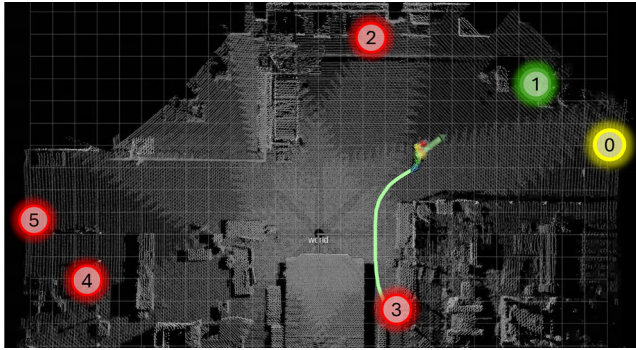


FIGURE 1. Visualisation of the proposed human trajectory classification (HTC) system in a factory setting. Predicted goal workstations based on partial human trajectory observations: Green (most likely), yellow (second most likely), red (unlikely). This enables proactive workstation preparation.

Human motion prediction (HMP) has rapidly evolved as a research area aimed at forecasting future human motion in complex and dynamic settings. In this paper, we focus specifically on human trajectory classification (HTC) in collaborative human-machine environments, a specialised subtask within the broader field of human trajectory prediction (HTP) [4]. While traditional HTP approaches predict continuous future positions along a path, HTC focuses on classifying the intended destination from partial trajectory observations, a critical capability for industrial applications where knowing the goal workstation enables proactive resource allocation and safety measures. Current research on HTP focuses mostly on applications like autonomous vehicles, service robots and advanced surveillance systems. On the other hand, it has disruptive potential in factories, an area where research on this topic is limited. HTC reduces the need for safety cages common in traditional industrial processes, thus increasing the usable shop floor by enabling hazardous machines to slow down before a human arrives. Additionally, by using classification techniques to predict the target workstation, it can increase productivity by proactively booting up collaborative workstations, saving time every time an operator enters the workstation to perform a task.

As shown in Fig. 1, this research proposes a predictive HTC system that dynamically estimates the operator's intended goal workstation. Different workstations are identified by a labelling number. As the operator moves closer to a workstation, the proposed algorithm gradually increases its confidence in predicting the operator's intended goal.

A. RELATED WORK

HTP has gained significant interest in recent years due to advancements in sensing technologies, reduced computational costs, and the growing trend of shared spaces between humans and machines. Fig. 2 shows how the topic has grown from a scant volume in 2010 to become a highly active research topic the last few years.

Traditional works often focus on pedestrian prediction using deterministic models [5], [6], [7]. These models rely on predefined equations to describe human behaviour, such as interactions between pedestrians, obstacles, and their environment. While effective in structured scenarios, these approaches are limited by their assumption of fixed rules and parameters. They struggle to capture the variability, randomness and dynamic nature of real-world human behaviour.

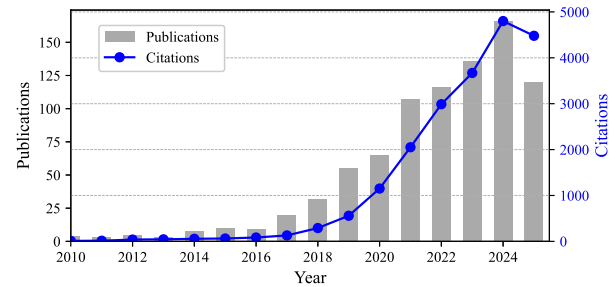


FIGURE 2. Publication and citation counts per year for the search word: "Human trajectory prediction" OR "Human motion prediction" OR "Human trajectory classification". This includes numbers from Web of Science Core Collection, Preprint Citation Index, MEDLINE, ProQuest Dissertations & Theses Citation Index and Grants Index. Extracted 17th of January 2026.

With the advancements in machine learning, there has been a shift towards data-driven methods for trajectory prediction [8], [9], [10], [11]. By leveraging real-world data, algorithms can adapt to diverse scenarios without relying on rigid assumptions. This enables a more robust approach that generalises better across environments.

A significant portion of research in this field has been dedicated to self-driving vehicles, where safety concerns are paramount when operating in human-populated environments [12]. A notable example is presented in [8], which employ a social-science long-short term memory network (SS-LSTM) to predict pedestrian trajectories in urban settings. Complementary to trajectory prediction, the use of deep recurrent neural networks (RNNs) to predict a human's 3D pose a few seconds into the future is demonstrated in [9].

The applications of HTP are diverse, and another field where it has caught interest is in the industrial domain. For example, a method to predict the goal locations of human operators walking through a warehouse using a hidden Markov model (HMM) is outlined in [10]. The model predicts human intentions based on decision patterns in a highly organised warehouse structure with shelves. However, in industrial environments with a more irregular or organic layout, this approach may be less effective, as human decision patterns might be more obscure. With the increasing focus on human-centric design principles in Industry 5.0, there is a growing opportunity to optimise industrial floors by integrating predictive systems that enhance collaboration and safety, while reducing reliance on rigid traditional layouts designed purely for separation.

Löcklin et al. demonstrated the use of schedule-based data for trajectory prediction in manufacturing environments [11]. They predict the destination of a worker based on current position of the worker, work schedule from the manufacturing execution system (MES) and a semantic map of the factory. However, their approach relies heavily on predefined schedules, limiting adaptability to real-time deviations. Reference [13] addresses this gap by only relying on real-time motion data, enabling more accurate predictions in dynamic settings and simplifying the adoption in different case studies by eliminating the need for MES-integration.

A challenge within HTP, especially in confined industrial spaces, is the compelling lack of relevant datasets, which was pointed out in [14]. This underpins the importance of finding new ways to gather enough data. Two promising techniques designed for situations with low data availability, is transfer learning and data augmentation. Transfer learning is a machine learning technique, where knowledge gained in one dataset is used to improve the model performance on another related dataset. Transfer learning is commonly used if the dataset at hand is too small, or to reduce the computational cost of training the model from scratch [15]. Data augmentation on the other hand, is the process of artificially generating new data from existing data. It is commonly used to improve model accuracy and reducing the risk of overfitting, especially in smaller datasets with a low diversity. For instance, in image processing, the same image can be used multiple times for training by applying transformations to the image, such as rotation, zoom, stretch, noise and blur. Timeseries data requires different transformations, such as flipping, down-sampling or adding slope or noise [16].

Research on data augmentation has advanced rapidly in recent years with generative adversarial networks (GAN) [17]. A GAN can generate synthetic data that resemble the statistical characteristics of training data, by simultaneously training two models that will play against each other: one generative model that tries to mimic the training data, and one model that tries to predict whether the generated data comes from the original dataset or from the generative model. A type of GAN that attend to temporal correlations unique to timeseries data is timeseries GAN (TimeGAN) [18]. Although GAN has proven to be useful in simulating human data [19], it requires a solid dataset as the foundation.

In industrial settings, trajectory data collection can be costly, time-consuming, or subject to safety constraints—factors that may limit the amount and quality of available data. When the dataset is too small or unrepresentative, TimeGAN may produce synthetic sequences that deviate from the true distribution of human trajectories, reducing the model's practical value. Moreover, high computational overhead is often involved in training deep learning models like TimeGAN, which can be a barrier for organisations with limited computational resources. Taken together, these

limitations highlight the importance of alternative methods, such as traditional stochastic methods.

An alternative to generating human trajectories was presented in our previous paper, where an Ornstein-Uhlenbeck (OU) process was applied [20]. The OU process is a well-known stochastic process introduced by Uhlenbeck and Ornstein in [21], where they describe a mathematically modified Brownian motion incorporating a drift toward a central point. For human trajectory simulations, it is used to model motion by combining randomness with a tendency to return to a mean path, effectively capturing realistic fluctuations while maintaining overall trajectory stability. While Gaussian process-based data augmentation have been explored in existing literature [22], the potential in using the OU-process for human motion data augmentation remains unexplored.

B. RESEARCH QUESTIONS AND CONTRIBUTIONS

Research on HTC for indoor industrial environments is limited. This paper aims to answer the following research questions:

- RQ1: What are the key challenges in predicting human trajectories, and how can they be addressed?
- RQ2: How can predictive HTC be implemented to enhance productivity and safety in industrial environments?
- RQ3: How can the performance of HTC models be improved in scenarios with limited data availability?

By addressing these research questions, this paper helps to close a current research gap with the following contributions:

- An end-to-end predictive HTC pipeline for indoor industrial environments, integrating LiDAR-based sensing, data processing, augmentation, and classification.
- A real-world case study from an actual industrial floor covering an area of 26 m × 20 m.
- A comparative analysis of data augmentation strategies (OU process vs. TimeGAN) for trajectory data in data-scarce scenarios, combined with transfer learning to improve classification performance.

Fig. 3 presents the proposed framework for improving HTC through data augmentation and transfer learning. The paper is organised to reflect the flow of this figure, starting with an explanation of the data collection procedure (Section II), followed by data processing (Section III). After data processing, the data is split into a training set and testing set. The training set is augmented to generate a synthetic dataset of trajectories as explained in Section IV. The section presents both TimeGAN- and OU-based generation, thus comparing a modern learning approach with a traditional statistical method. Section V presents the proposed LSTM model with the corresponding training procedure, which is followed by results and discussions in Section VI before the paper is summarised and concluded in Section VII.

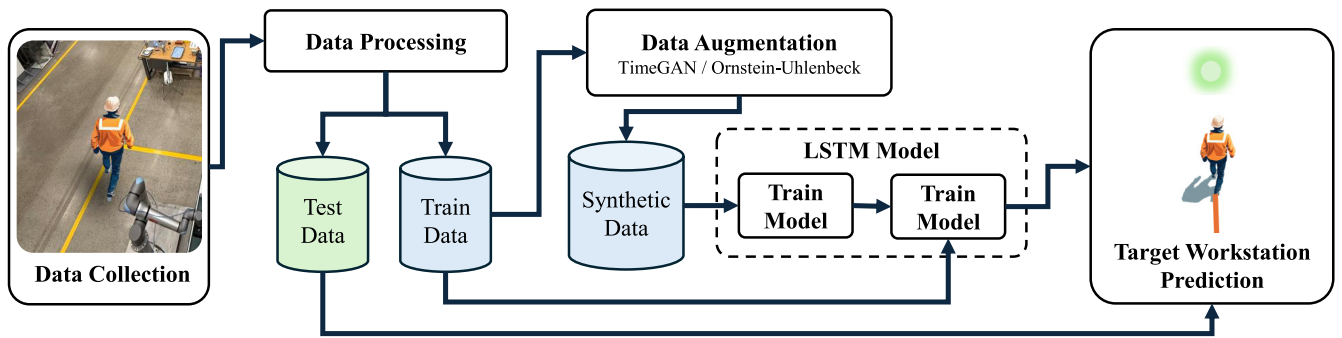


FIGURE 3. Proposed framework for improving human trajectory classification (HTC) through data augmentation and transfer learning. It starts on the left with collecting real-world trajectory data. This data is processed and split into a train set and a test set. Further on, the data is augmented using the stochastic Ornstein-Uhlenbeck process. This results in a synthetic dataset, which is used to train the model. Further on, the model is re-trained on real-world data, before it is validated on the test data.

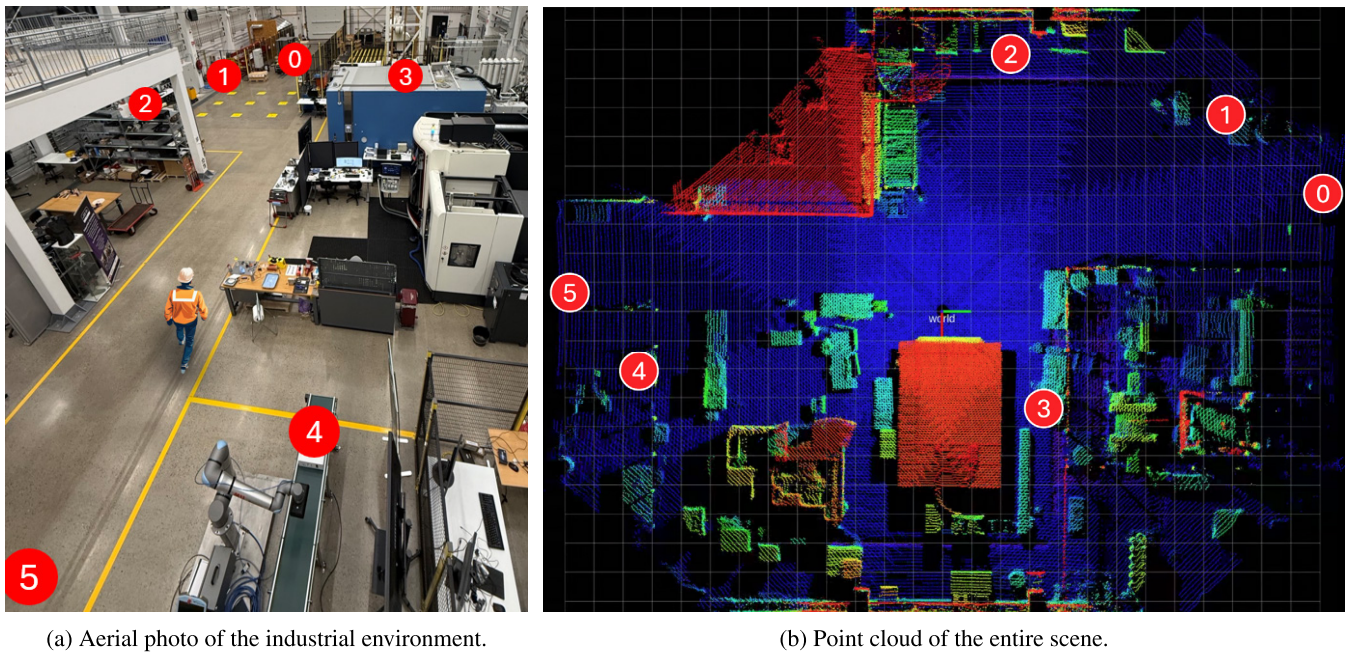


FIGURE 4. Overview of the industrial environment used in the case study including: (a) an aerial photo of the industrial environment and (b) a point cloud of the industrial environment captured from a LiDAR mounted to the roof, covering an area of 26 m × 20 m. The colours indicate the z-coordinate, representing the height, where blue is the floor, while the highest points are red. The red, numbered circles represent the different workstations used in the experiment.

II. DATA COLLECTION

This section explains the details regarding the experimental setup employed to collect trajectory data, including sensor and hardware configuration, human detection, and the data collection procedure.

A. EXPERIMENTAL CASE STUDY

A real-world case study is investigated to obtain realistic data and validate the performance of the proposed prediction framework. Fig. 4a shows a photo of the industrial floor, where the red circles represent the workstation configuration. Additionally, a point cloud of the entire scene can be seen in Fig. 4b, which covers an area of 26 m × 20 m. The workstation configuration seen in these

figures reflects an organic industrial layout, which is not constrained by traditional safety cages and rigid designs. This flexibility aligns with the principles of Industry 5.0, where adaptable, human-centric, and collaborative environments are prioritised, making our case study highly relevant for the factories of the future.

B. NOVEL SENSOR AND HARDWARE CONFIGURATION

The dataset is based on a state-of-the-art, high-performance LiDAR system, developed from research presented in [23]. As shown in Fig. 5, this system features an innovative design with eight distinct mirror segments, each inclined at 34 degrees. This design enhances the embedded LiDAR's field of view (FOV) and resolution by effectively

redistributing its original $360^\circ \times 45^\circ$ FOV into a more focused $90^\circ \times 90^\circ$ directional FOV. In this work, the system serves as an optimal spatiotemporal data acquisition platform for real-time human detection from a bird's-eye view.

The embedded LiDAR model used in the setup is the OS1-128, developed by Ouster.¹ The OS1-128 features 128 laser emitters, uniformly distributed across a 45-degree range, producing dense point clouds with up to 128×4096 data points per frame.

For dataset generation, a configuration of 128×2048 data points per frame was selected, operating at a steady 10 frames per second. This setup balances high resolution with frame rate, enabling dynamic environmental sensing and making it well-suited for applications requiring precise, real-time perception. The resulting point cloud, shown in Fig. 4b, provides a high-resolution representation of the environment, capturing fine details essential for accurate detection and analysis.



FIGURE 5. The LiDAR system reflects the entire FOV of the rotating LiDAR towards the scene, expanding the field of view, while maintaining the range and robustness of the rotating LiDAR. (© Atle Aalerud, 2024. Picture may be reproduced with attribution.)

C. HUMAN DETECTION AND TRACKING

For human detection, we apply the method proposed by Licaret et al. [24], which employs a multi-stage training pipeline leveraging synthetic and real-world data. In the first step, synthetic data is generated using a customised Super-Resolution LiDAR simulation that replicates the industrial environment by populating the scanned space with 3D human models. The second step applies domain adaptation techniques, introducing effects such as ray drop, range noise, and mirror edge drop to align synthetic point clouds with real-world characteristics, improving the model's generalisation. An initial model is trained on this dataset to detect personnel in point cloud data. Finally, a self-supervised learning approach is used, where the initial model pseudo-labels real, unannotated data, which is then combined with synthetic data for retraining. This iterative process reduces domain discrepancies and enhances detection accuracy, making

¹ More details are available at <https://data.ouster.io/downloads/datasheets/datasheet-rev7-v3p1-os1.pdf>, accessed May 2024.

the model robust for human-robot interaction in industrial settings.

Our personnel tracking system, as illustrated in Fig. 6, uses an augmented Kalman filter to monitor and track human movement within the workspace in real-time, as described by Aalerud and Hovland [25]. The detections are fed into the Kalman filter, which groups overlapping detections (in the case of multiple sensors) and assigns unique IDs to individuals in the area. The filter is designed to handle occlusions and improve accuracy by dynamically adjusting its parameters based on available sensor data, reducing noise and smoothing velocity estimates.

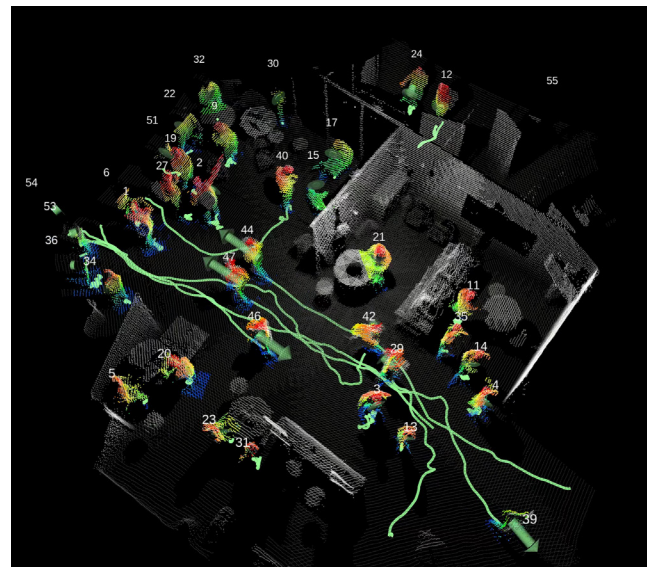


FIGURE 6. Detection, tracking, and prediction of persons illustrated using a dataset recorded at the European Robotics Forum in Rotterdam, 2022. Detected persons are represented as colored point clouds, with person IDs displayed above them. Historic tracks are shown in green, while velocity vectors are visualised as green arrows. (© Atle Aalerud, 2022. Picture may be reproduced with attribution.)

D. DATA COLLECTION PROCEDURE

To collect trajectory data, an experiment involving nine volunteer participants is designed, each walking for 20 minutes within the industrial floor. Participants are tasked with moving between six predefined workstations, simulating natural movement patterns. To ensure randomisation of trajectories, a dice roll is used to assign their next target workstation, reducing potential bias in trajectory selection. Experience, culture and age varied amongst the group, representing five different nationalities and a mean age of 32 with a standard deviation of six.

To maintain engagement during the 20-minute session, participants completed a simple quiz with questions at each workstation, allowing them to stay focused yet entertained without overthinking their movement patterns. This setup ensured that the trajectories represented natural walking behaviour while maintaining a controlled environment suitable for data analysis. All participants provided informed

consent before taking part in the study, having been briefed on the purpose, data usage, and their right to withdraw at any time. The resulting dataset captured a diverse range of trajectories, including all possible workstation transitions with varied walking speeds.

III. DATA PROCESSING

In this section, we outline the procedures for transforming the trajectory data into a clean and well-structured dataset. First, we apply filters to eliminate spurious or low-quality points. Next, we label each data point with trajectory identifiers and workstation information. Finally, we prepare the labelled data for machine learning by normalising, cropping and reshaping the trajectories, ensuring compatibility with the predictive model.

A. FILTERING RAW DATA

Before labelling the trajectories several filters are applied to remove spurious, low-quality and duplicate points.

- 1) $velocity_scalar > 0.1$: Exclude rows where the velocity is close to zero, as these often correspond to sensor noise or detections that do not belong to actual walking trajectories.
- 2) Remove undesired bounding box regions to mitigate false positive human detection data. For instance, a robot in the scene may be perceived as a human at certain times. Removing all points in close proximity to the robot can eliminate all these false positives.
- 3) Drop consecutive duplicates in x- and y-coordinates which are deemed artifacts, or bugs, in the raw data, i.e., $x_{t-1} = x_t$ and $y_{t-1} = y_t$.
- 4) Manually remove detections of people that are not part of the experiment.

B. LABELLING THE DATASET

First, each data point is labelled with the workstation it falls within. A workstation is defined by a 2.5 m $\times$ 2.5 m bounding box around its central point. If the data point does not fall within a workstation, the workstation field is labelled as -1 , and means that this point is likely part of a transition between workstations.

Next, each data point is assigned a trajectory identifier (`trajectory_id`) which increases whenever a new trajectory begins. That means, a participant has thrown the dice and started walking to the next workstation. This scenario is automatically detected by checking the following conditions:

- 1) The person is within a workstation, and the orientation changes with more than $\Delta\theta_{thres}$ degrees between two consecutive data points, as described by the following equation:

$$\min(|\theta_{t-1} - \theta_t|, 360^\circ - |\theta_{t-1} - \theta_t|) > \Delta\theta_{thres}. \quad (1)$$

- 2) The person has remained within a workstation, moving less than 10 cm over a period of more than 1 second. This means that the person has likely stopped at a

workstation, and the following points will belong to the next trajectory.

After this process, each data point will have a `trajectory_id`. Additionally, each trajectory will start within one workstation, and end up in another. The final step of the labelling process, is to identify the workstation of the last data point of a trajectory, and assign it as the goal of each data point sharing the same `trajectory_id`. This goal information will later be used for classification.

C. PREPROCESSING FOR TRAJECTORY LEARNING

Preparing the dataset for machine learning through preprocessing is a crucial step that ensures compatibility with the chosen algorithm, reduces computational cost and improves model performance. The following steps are applied to the dataset before the learning process:

1) SHUFFLE TRAJECTORIES

To ensure a mix of participants in both the training and test sets, the trajectories are shuffled based on their unique identifiers. This prevents overfitting to specific individuals and ensures diverse training data.

2) CROP THE DATA TO EXCLUDE THE LAST ΔT SECONDS

Since only the most relevant trajectory segments should be used, trajectories are truncated by removing the last few seconds. Let t_{end} be the timestamp of the last recorded data point in a trajectory, and let ΔT be the cropping duration. The retained portion of the trajectory is:

$$\mathcal{T}_{cropped} = \{(x_t, y_t) : t \leq t_{end} - \Delta T\}. \quad (2)$$

For the simulated data, which is generated as explained in Section IV, the time axis was not realistic. Thus, a percentage-based crop was applied, retaining a random 65%–100% segment of each trajectory. Let n be the total number of points in a trajectory and let r be a random variable uniformly distributed between 0 and 1, i.e. $r \sim U(0, 1)$. The cropped portion is:

$$\mathcal{T}_{cropped} = \{(x_i, y_i) : i \leq \lfloor n \times (0.65 + 0.35r) \rfloor\}. \quad (3)$$

This ensures that the model learns incomplete trajectories making it more robust in classifying trajectories a few seconds before arriving at a workstation.

3) CHOOSE A FIXED NUMBER OF POINTS

To balance between preserving motion characteristics and maintaining computational efficiency, a fixed number of 30 points is chosen for each trajectory.

4) ENSURE A FIXED NUMBER OF POINTS PER TRAJECTORY

Since different trajectories have varying lengths, they are either downsampled if they contain too many points, or interpolated if they have too few. The method applied depends on the number of recorded points, N , relative to the desired sequence length, L .

- *Downsampling* (if $N > L$): The new points are selected by choosing L evenly spaced indices using:

$$i_k = \left\lfloor k \cdot \frac{N-1}{L-1} \right\rfloor, \quad k = 0, 1, \dots, L-1. \quad (4)$$

- *Interpolation* (if $N < L$): Linear interpolation is performed to estimate missing values:

$$x_{t_k} = x_{t_i} + \frac{t_k - t_i}{t_{i+1} - t_i} (x_{t_{i+1}} - x_{t_i}), \quad (5)$$

where $t_i \leq t_k \leq t_{i+1}$ are the two nearest known timestamps, and a similar equation is applied for y_{t_k} .

5) NORMALIZATION

To ensure numerical stability and prevent dominance of certain features due to scale differences, min-max normalisation is applied to all trajectory features, transforming the values to span from 0 to 1:

$$X_{\text{norm}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}}, \quad (6)$$

where X is the original feature value, $X_{\min}$ and $X_{\max}$ are the minimum and maximum values in the dataset, and X_{norm} is the normalised value.

After these preprocessing steps, the dataset is well-structured and ready for data augmentation, and trajectory learning.

IV. DATA AUGMENTATION

Collecting large volumes of human trajectory data is both time consuming and tedious for participants. It can also be expensive if you have to close certain areas to obtain a clean, high quality dataset from a safely controlled area. This is therefore a great use case for transfer learning and data augmentation. This section shows how TimeGAN is used to generate synthetic data and introduces a new timeseries augmentation method using the OU process.

A. GENERATING HUMAN TRAJECTORIES USING TIMEGAN

Since human trajectories inherently involve complex spatiotemporal patterns, e.g. walking speeds, pauses, and turns, TimeGAN represents a promising, state-of-the-art alternative that can generate a varied and realistic synthetic dataset.

To ensure that TimeGAN learns meaningful patterns, the dataset is split into groups based on trajectories with the same start-goal pair. This allows the model to focus on learning the nuances of individual trajectory segments rather than attempting to generalise across different movement patterns at once. By training on each subset separately, the synthetic trajectories produced by TimeGAN maintain consistency with realistic transitions between workstations, improving the overall quality of generated data. The training set includes only 192 trajectories. Divided on the total number of start-goal pairs with 6 workstations ($6 \times 5 = 30$), that results in an average of 6.4 trajectories per start-goal pair, which can be considered a small dataset in a machine learning setting.

A series of hyperparameter testing decides the final configuration for TimeGAN. Specifically, a batch size of 32, a learning rate of 1×10^{-3} with Adam betas (0.9, 0.999), a hidden dimension of 24, a feed-forward layer size of 128, a noise dimension of 32, and $\gamma = 1$ is selected. A crucial parameter for TimeGAN is the number of training steps. Generally, the higher the train steps, the better and more robust the model becomes. However, too many steps may result in overfitting, making it less realistic in capturing unseen data. This is especially important in cases with small datasets, where it is necessary to move beyond the limited samples available. Number of training steps is decided through testing within a range of 100-5000 to find the tradeoff between underfitting and overfitting. The result is validated by inspecting the synthetically generated data, overlaid on the real-world dataset, where the most suitable number of training steps for this use case is found to be 200. This relatively low number of training steps is found optimal due to the low amount of training data, as excessive training on a small dataset increases the risk of overfitting.

B. GENERATING HUMAN TRAJECTORIES USING THE ORNSTEIN-UHLENBECK PROCESS

Due to the limited amount of available training data, the OU-process is used to simulate additional trajectories. These simulations are done based on the existing data by first defining reference paths based on polynomial regression. Outliers in the existing data are first removed in order to generate the most suitable reference paths. An outlier is typically a trajectory where a person makes a mistake by walking in the wrong direction, moving outside the designated area, or suddenly turning. The polynomials are given by the following equations:

$$\begin{aligned} \hat{x}(t) &= \beta_0^{(x)} + \beta_1^{(x)} t + \beta_2^{(x)} t^2 + \dots + \beta_d^{(x)} t^d, \\ \hat{y}(t) &= \beta_0^{(y)} + \beta_1^{(y)} t + \beta_2^{(y)} t^2 + \dots + \beta_d^{(y)} t^d, \end{aligned} \quad (7)$$

where $t \in [0, 1]$ is the normalised path parameter (cumulative distance divided by total distance), while $\beta_k^{(x)}$ and $\beta_k^{(y)}$ are the polynomial coefficients, estimated by ordinary least squares for x and y respectively. d is the chosen polynomial degree, which is selected by manually inspecting the resulting graph overlaid on the real dataset.

Furthermore, the OU process is used to generate the trajectories, using equations for a numerical OU process, which is given below:

$$dx_t = -\theta (x_t - x_{\text{ref}}) dt + \sigma dW_t, \quad (8)$$

where x_t is the simulated variable. θ controls the intensity of the mean reversion, pulling the simulated value towards the reference point x_{ref} . Further on, dt is the time step, while σ represents the volatility parameter that controls the level of randomness in the simulation and dW_t is a Wiener process.

Using the OU process to model position directly leads to erratic behaviour unsuitable for human walking dynamics, despite fine-tuning the parameters θ and σ . However, when

the process models velocity instead of position, the integrated trajectories yield smoother, more realistic movements. The reference velocity can be obtained from the discrete derivative of the regression-based reference trajectories from Eq. 7. This velocity-based approach introduces a cumulative drift, as the velocity error is not corrected over time. Therefore, a hybrid approach is applied where velocity follows an OU process, supplemented by a corrective term tied to positional errors. The final implementation is presented in Eq. 9, which incorporates the position correction:

$$\begin{aligned} \vec{v}(t) = & \vec{v}(t - dt) - \theta_v \cdot (\vec{v}(t - dt) - \vec{v}_{ref}(t)) \cdot dt \\ & + \sigma \cdot dW_t - \theta_p \cdot (\vec{p}(t - dt) - \vec{p}_{ref}(t)) \cdot dt, \end{aligned} \quad (9)$$

where $\vec{v}$ is the simulated velocity vector in 2D, t is time, $\vec{v}_{ref}$ is the reference velocity, $\vec{p}$ is the position vector, and $\vec{p}_{ref}$ is the reference position. This equation smoothens the simulated position trajectory by modelling the velocity as an OU process, while mitigating drift via positional correction.

To ensure convergence to the workstation, θ is defined to increase as the person nears the workstation, thus increasing the reversion strength to the reference path. The equations below shows how the mean reversion strength for velocity and position is defined:

$$\theta_v(n) = 1.5 + 0.8 \frac{n}{N}, \quad (10)$$

$$\theta_p(n) = 0.3 + 2.9 \left(\frac{n}{N} \right)^5, \quad (11)$$

where $\frac{n}{N}$ is a progression indicator ranging between 0 and 1, where n is the current index along the reference path, while N is the total number of points on the discrete reference path. The model uses $\sigma = 0.15$.

The reference index n is determined by finding the point on the reference path with the smallest Euclidean distance to the simulated position. Fig. 7 illustrates the search process, where the green region indicates the search segment at time t . At each step, the associated velocity and position values are fed into the OU equations. As n increases, the search segment dynamically shifts along the reference path, keeping the computational effort efficient by only considering a fixed number of 100 points in each step.

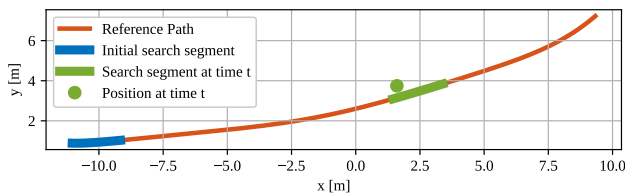


FIGURE 7. The search algorithm to find nearest point on the reference path [20].

Noise and sampling frequency are applied to improve the realism of the data. A random number within ± 0.01 m is added and the values are only stored when a certain sampling time is reached.

V. PREDICTIVE HUMAN TRAJECTORY CLASSIFICATION

The processed datasets gathered through experiments and simulations provides the foundations for the next step, which is to predict the trajectory goal ahead of time. This is a classification problem that takes spatiotemporal data as input and predicts the probability associated with each workstation being the intended goal.

Algorithms like Gaussian naive Bayes, support vector machine, and random forest are commonly used for learning based classifications. However, these conventional algorithms do not intrinsically model the temporal dependencies available in trajectory data. Recurrent neural networks (RNN) address this limitation by taking a sequence of data points as input. It can learn how different motion patterns relate to different human intentions, thus providing a better data foundation for predicting the future of a trajectory. An RNN in its simplest form, relies on recurrent connections in the hidden layer but often suffer from vanishing or exploding gradients over long sequences [26]. A gated recurrent unit (GRU) is one improvement, using update and reset gates to regulate information flow and mitigate gradient issues. A more powerful type of RNN is the LSTM network. It includes an input gate, output gate and forget gate in addition to a cell state that preserves long-range dependencies more effectively, as shown in Fig. 8.

In this study, a stacked LSTM architecture is employed, consisting of three LSTM units followed by two plain artificial neural network (ANN) layers. To ensure stable training and improve performance, batch normalisation is applied between each layer, normalising the data before forwarding it to the next stage.

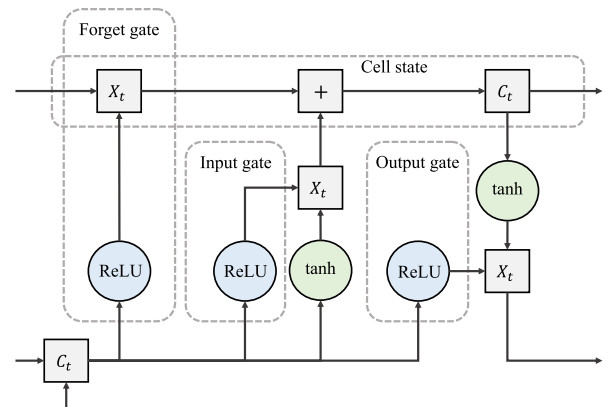


FIGURE 8. The general structure of a long-short term memory (LSTM) network. The cell state preserves long-term dependencies, regulated by the input, output, and forget gates [20].

A. HYPERPARAMETER TUNING

Finding the optimal hyperparameters of a neural network is a crucial prerequisite to optimise model performance. This can be done by setting threshold values and step size, and checking all combinations within the thresholds. This method is called grid search, and ensures a thorough exploration

of different parameter combinations. However, the number of combinations increases exponentially as the number of hyperparameters rise, making it computationally expensive.

To identify the most suitable hyperparameters for the stacked LSTM model designed in this study, a random search algorithm is used. It checks random combinations of hyperparameters without checking all possible combinations. While having a much lower computational cost, it also comes with a risk that there might be better combinations that remains unexplored [27]. The algorithm is employed to run 10 tests with different hyperparameter combinations, randomly selected within a predefined range. Combinations are tested through a training sequence of 10 epochs using the simulated data, where the combination that achieves the best validation accuracy is selected for further training.

To ensure an effective random search, limits are set on each hyperparameter to keep the search space relevant and computationally feasible. The number of LSTM units is constrained between 100 and 400 for the first layer, 50 and 300 for the second, and 50 and 200 for the third, all with a step size of 50. The dense layer units range from 100 to 300, and the optimiser is selected from *adam*, *sgd*, and *rmsprop*. These bounds prevent extreme values that could hinder convergence while maintaining sufficient variability to find an optimal configuration. The selected hyperparameters are summarised in Table 1, including the most important training configuration parameters.

TABLE 1. Network architecture and key hyperparameters for the stacked long short-term memory (LSTM) network.

Layer	Hyperparameter	Value
LSTM 1	Units	200
	Activation function	tanh
LSTM 2	Units	100
	Activation function	tanh
LSTM 3	Units	50
	Activation function	tanh
Dense 1	Units	100
	Activation function	relu
Dense 2 (Output)	Units	6
	Activation function	softmax
Training Config	Optimiser	adam
	Number of epochs	200
	Learning rate at epoch < 100	0.001
	Learning rate at epoch ≥ 100	0.0001
	Batch size	16

B. MODEL TRAINING PROCEDURE

The hyperparameters forms the foundation for the training process. To highlight the effect of different data sources, three main scenarios are examined:

Scenario 1: In the first scenario, the model undergoes traditional training and validation on the real-world dataset. The data is split into 40% for training and 60% for testing. This relatively low amount of data for training is chosen intentionally to emphasise the challenges of limited data availability when collecting data of human subjects.

Scenario 2: For the next scenario, the model is only trained on synthetic data, with the data split into 75% for training and 25% for testing. Further on, it is validated on the real-world dataset.

Scenario 2.1: The synthetic data is generated using TimeGAN. This synthetic dataset is shown in Fig. 9b.

Scenario 2.2: Synthetic data is simulated as an OU process, resulting in the dataset shown in Fig. 9c

Scenario 3: In the third scenario, the model benefits from a combination of synthetic and real-world data. It loads the trained model from Scenario 2 and freezes the first LSTM layer. This means that the weights of the first layer will not be updated, thereby preserving robustness from extensive amounts of synthetic data. The remaining LSTM layers and dense layers are still trainable and are updated to fit the real-world dataset. The real-world data is divided into the same train-test split as Scenario 2: 40% for training and 60% for testing.

Scenario 3.1: Trains the pre-trained model from Scenario 2.1 with real-world data.

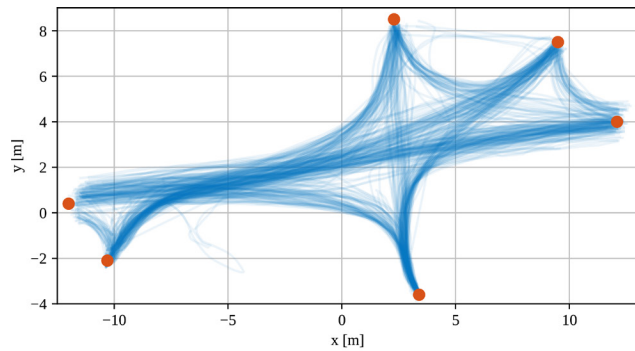
Scenario 3.2: Trains the pre-trained model from Scenario 2.2 with real-world data.

VI. RESULTS AND DISCUSSIONS

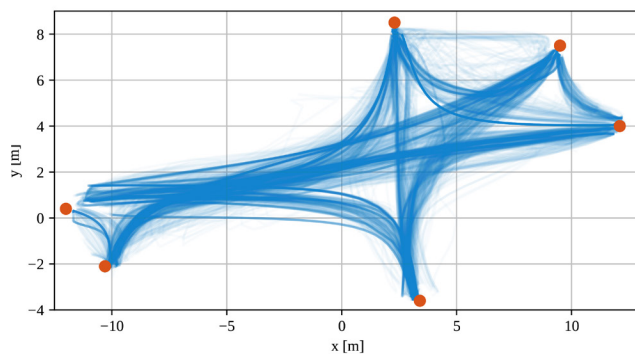
Fig. 9 shows the trajectory data collected from simulations and real-world experiments. The simulated datasets includes 3000 distinct trajectories, while the real-world dataset includes 487, each reflecting randomly selected start-goal pairs among the available workstations. After splitting the data into training and testing sets (75% training / 25% testing for simulation data and 40% training / 60% testing for real-world data), 2250 synthetic and 192 real-world trajectory samples remain for training.

The results from the five considered scenarios are summarised in Fig. 10. This chart presents the results from classifying the goal workstation at different time horizons spanning from 1 second to 5 seconds before the person arrives at the workstation. It compares the five training scenarios described in Section V-B. The algorithm trained exclusively on real data (Sce. 1) performs better than the ones trained only on simulation data (Sce. 2.1 and Sce. 2.2) in all prediction horizons, while the hybrid training strategy (Sce. 3.1 and Sce. 3.2) resulted in notable performance gains, particularly for longer time horizons. For predictions 3 seconds ahead, the classification accuracy improved from 87.5% in Sce. 1 to 94.9% in Sce. 3.1 and 95.6% in Sce. 3.2. Table 2 complements this comparison chart by displaying evaluation metrics for all scenarios at different time horizons.

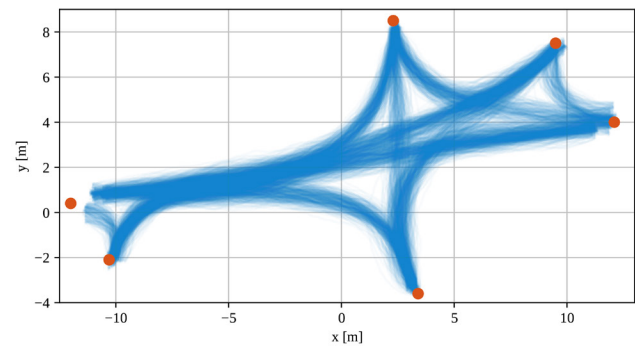
Fig. 11 presents the confusion matrices for scenario 1, 2.2 and 3.2 when classifying 4 seconds ahead. These matrices clearly show that the most challenging workstations to predict are number 4 and 5. This performance variation across workstations can be attributed to the physical layout of the industrial floor. Workstations 4 and 5 are located behind a shared obstacle, and trajectories heading toward either one remain largely overlapping until the operator passes



(a) Real-world trajectory data.



(b) Simulated trajectory data using TimeGAN.



(c) Simulated trajectory data using Ornstein-Uhlenbeck.

FIGURE 9. Heatmaps of trajectory data: (a) Real-world data; (b) Synthetic data based on TimeGAN; (c) Synthetic data based on the Ornstein-Uhlenbeck process.

this divergence point. Consequently, at longer prediction horizons, the partial trajectory does not yet contain discriminative spatial features to distinguish between these two destinations. In contrast, workstations such as 3 and 6, which are reached via more spatially distinct paths, are classified with consistently higher accuracy. This is further evident in Fig. 12, where Trajectory 5 is a wrongly classified example: the operator has yet to pass the last obstacle and shows no clear directional indication of the intended workstation. Trajectories 3 and 4 demonstrate the model's robustness to minor detours by correctly classifying despite initial movement in the wrong direction.

Other examples are outlined in Trajectory 0-4. Interestingly, it is able to predict the correct workstation with a

TABLE 2. Metrics (Accuracy, Precision, Recall, F1-score) for each scenario (Scenario 1, 2.1, 2.2, 3.1, 3.2) across different time horizons (1s–5s), where the time horizon represents the prediction interval into the future. The bolding is determined by the highest value in each metric.

Training Scenario	Metric	1s	2s	3s	4s	5s
Scenario 1 Trained on Real Data	Accuracy	0.98	0.98	0.88	0.80	0.75
	Precision	0.98	0.98	0.88	0.81	0.75
	Recall	0.98	0.98	0.88	0.80	0.75
	F1-score	0.98	0.98	0.88	0.80	0.75
Scenario 2.1 Trained on TimeGAN Data	Accuracy	0.96	0.91	0.85	0.78	0.63
	Precision	0.96	0.91	0.85	0.79	0.67
	Recall	0.96	0.91	0.85	0.78	0.63
	F1-score	0.96	0.90	0.85	0.78	0.63
Scenario 2.2 Trained on OU Data	Accuracy	0.94	0.88	0.79	0.75	0.61
	Precision	0.95	0.89	0.80	0.76	0.62
	Recall	0.94	0.88	0.79	0.75	0.61
	F1-score	0.94	0.88	0.78	0.75	0.61
Scenario 3.1 Trained on TimeGAN + Real Data	Accuracy	0.98	0.96	0.95	0.87	0.82
	Precision	0.98	0.97	0.95	0.87	0.82
	Recall	0.98	0.96	0.95	0.87	0.82
	F1-score	0.98	0.96	0.95	0.87	0.81
Scenario 3.2 Trained on OU + Real Data	Accuracy	0.99	0.98	0.96	0.88	0.81
	Precision	0.99	0.98	0.96	0.88	0.82
	Recall	0.99	0.98	0.96	0.88	0.81
	F1-score	0.99	0.98	0.96	0.88	0.81

high confidence in Trajectory 3 and 4, where both show a sequence of a person walking in the wrong direction in the beginning. In Trajectory 3, the person initially appears to be heading toward workstation 4 or 5 but ultimately turns toward workstation 3. Similarly, Trajectory 4 includes a detour to a non-active area between workstation 3 and 4, before finding the right way again. This demonstrates the algorithm's robustness in managing minor deviations from expected paths and effectively handling small human navigation errors.

A. INDUSTRIAL SAFETY IMPLICATIONS

In real industrial deployments, prediction errors carry asymmetric costs. A false negative (i.e. failing to predict the correct workstation) may delay proactive safety measures such as slowing down hazardous machinery, posing direct risks to operator safety as outlined in ISO 15066 [28]. On the other hand, a false positive (i.e. predicting the wrong workstation) may trigger unnecessary machine slowdowns, resulting in productivity loss, equipment wear, and increased energy consumption [29]. This asymmetry implies that recall (i.e., correctly identifying the true destination) is the more safety-critical metric, while precision relates more directly to productivity. As shown in Table 2, the best-performing scenario (Sce. 3.2) achieves a recall of 0.88 at the 4-second horizon, meaning that in 88% of cases the system correctly identifies the destination with sufficient lead time for safety interventions. The remaining 12% primarily involve confusion between adjacent workstations 4 and 5 (see Fig. 11), where the physical layout inherently limits early discrimination. In a deployed system, this can be mitigated through conservative safety policies that apply protective

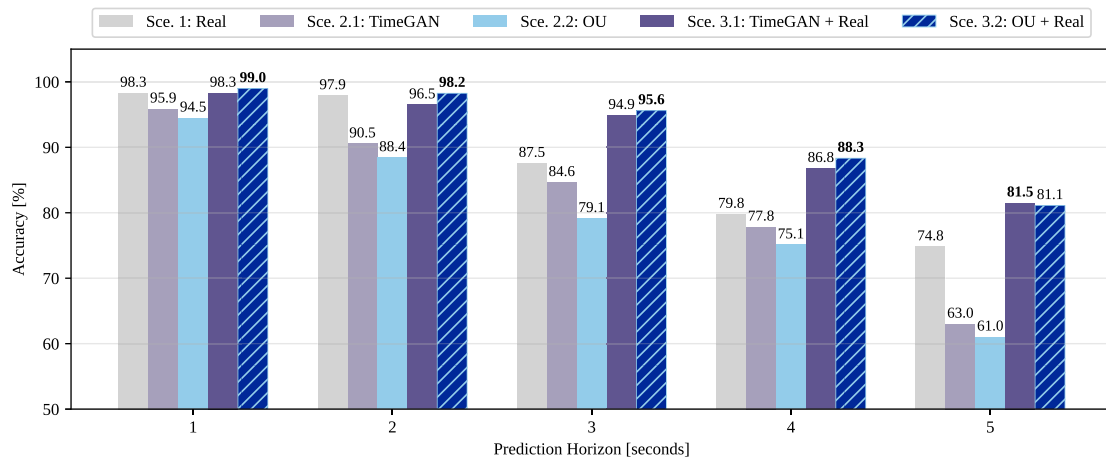


FIGURE 10. Prediction accuracy at different prediction horizons (1s to 5s) under five training scenarios: (1) training on real data alone, (2.1) training on synthetic data based on TimeGAN, (2.2) training on synthetic data simulated as an Ornstein-Uhlenbeck (OU) process, (3.1) training on TimeGAN data, and then on real data, and (3.2) training on OU data, and on real data.



FIGURE 11. Confusion matrices illustrating model performance on predicting the target workstation 4 seconds in advance on the real-world dataset across different training scenarios: (a) Training on real data; (b) Training on Ornstein-Uhlenbeck (OU)-based synthetic data; (c) Training on OU-based synthetic data, then using transfer learning to further train it on on real-world data.

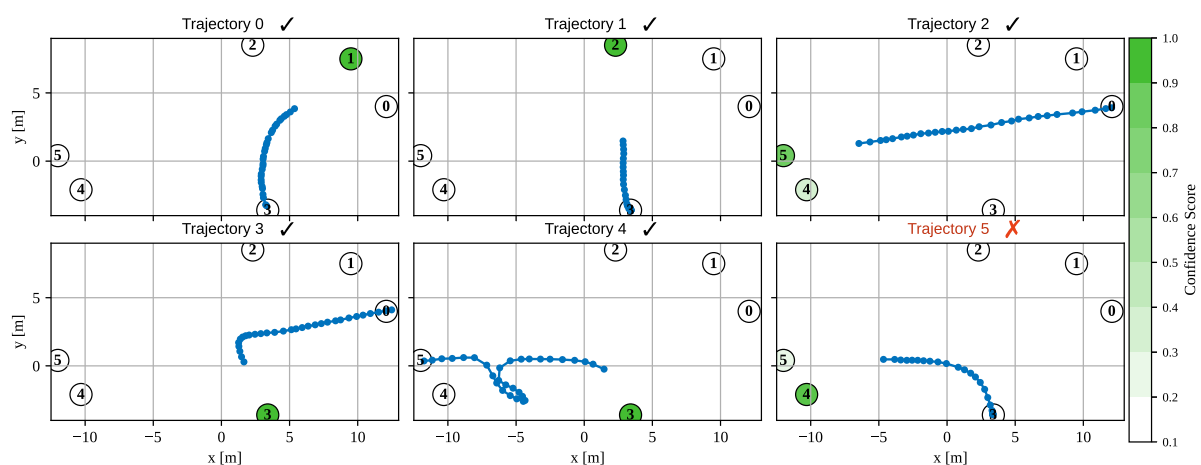


FIGURE 12. A selection of trajectories missing the last four seconds. The goal workstation is classified using the stacked LSTM model trained on Ornstein-Uhlenbeck (OU)-based synthetic data and real-world data (Scenario 3.2). The plots show examples of correctly classified trajectories (Trajectory 0-4) as well as one wrongly classified trajectory (Trajectory 5).

measures when the model assigns comparable probabilities to multiple neighbouring workstations.

Table 3 illustrates this asymmetry through a conceptual decision outcome matrix. Consider a scenario where the

model must decide 4 seconds before arrival whether to reduce robot speed at a given workstation. A false negative (bottom left cell) means the operator arrives at a workstation where the robot is still operating at full speed, representing an irreversible safety failure. A false positive (top right cell) means the robot slows down unnecessarily but resumes normal operation after a brief timeout, representing a recoverable productivity cost. Over an 8-hour shift with hundreds of transitions, accumulated false positives carry a non-trivial productivity penalty, yet each individual event is fully recoverable. A single false negative, by contrast, may result in an irreversible injury. This fundamental difference in severity is why recall should be weighted more heavily than precision when evaluating trajectory classification for safety-critical deployment.

TABLE 3. Conceptual decision outcome matrix illustrating asymmetric misclassification costs in predictive HTC. Each cell shows the operational consequence when the predicted destination (row) is compared against the actual destination (column).

		Actual	
		WS_i	Not WS_i
Predicted	WS_i	True Positive Robot slows at WS_i → Safe and correct	False Positive Robot slows unnecessarily → Productivity loss
	Not WS_i	False Negative Robot at full speed → Safety risk	True Negative Robot at full speed → Efficient and correct

B. ANSWERS TO RESEARCH QUESTIONS

1) RQ1

There are several challenges with predicting human trajectories, some of which is seen in existing literature, in the results of this paper and the presented case study. Collecting and processing data from the experimental setup proved to be a time consuming task, which represents a bottleneck in the development of human-centric industrial systems. Additionally, there are ethical concerns, especially related to privacy when collecting data of human participants [30]. By utilising a relatively low-resolution LiDAR, the privacy concern is mitigated by anonymising the collected data. Robustness and generalisability over multiple domains is another challenge that proves difficult. Although the proposed model architecture might suit multiple use cases, it requires specific data from the environment for use-case-specific training.

2) RQ2

Through the experimental setup, this paper demonstrates the possibility to enhance productivity and safety in industrial settings. By classifying target workstations of operators in real-time, the predictive HTC system can be used to proactively boot up collaborative machines or slow down hazardous machines, thereby decreasing stall time.

The International Organization for Standardization (ISO) suggests, in ISO 15066:2016 [28], to use sensors for detecting a human in the vicinity of the robot to trigger a protective stop, and regulating the speed and force when in close proximity to humans to mitigate collision risks. However, the standard does not address motion prediction as a means to enhance risk assessment [12]. By forecasting human movements, robotic systems can proactively adapt their actions, leading to smoother interactions and reducing the likelihood of hazardous encounters. Moreover, this approach can help minimise the reliance on safety cages, which are often space-inefficient and hinder the implementation of human-centric design principles central to Industry 5.0.

3) RQ3

This paper explores methods to handle data-scarcity through the use of data augmentation and transfer learning. By comparing learning based methods like TimeGAN with stochastic methods through OU-based simulations, our study shows different approaches with their benefits and drawbacks. The results suggest the smaller the dataset, the more relevant are traditional stochastic methods. However, learning based methods like GAN are still easier to implement with higher dimensional data, as they capture more complex patterns, and will often perform better with larger, more complex datasets.

C. FUTURE WORK

Future work will focus on publishing the dataset, including segmented human point clouds and corresponding trajectory data. The classification approach can be extended to incorporate planning-based approaches to estimate future movement, enabling predictive planning for mobile robots. Additionally, the methodology can be expanded to 3D, enhancing usability in other use cases, for instance with stairs. The transfer learning approach leveraging OU-based simulations can also be used to address challenges in HMP with limited datasets, such as for tracking arm movements in close proximity to robots.

VII. CONCLUSION

This study introduces a novel approach for predictive human trajectory classification (HTC) in industrial environments, focusing on overcoming the challenge of limited real-world data. We employ a stacked long-short term memory (LSTM) network to classify human trajectory destinations with high accuracy. The model was trained to classify the target workstation of the human operator from partial trajectory observations, achieving robust performance even in scenarios with limited data. The paper compares 5 training scenarios: (1) Training on real data; (2.1) Training on TimeGAN-based synthetic data; (2.2) Training on Ornstein-Uhlenbeck (OU)-based synthetic data; (3.1) Training on TimeGAN-based synthetic data as well as real data; (3.2) Training on OU-based synthetic data as well as real data.

The results demonstrate that although generative adversarial networks (GANs) offer promising capabilities in data

generation, they can be outperformed by more traditional statistical models, such as the Ornstein-Uhlenbeck (OU) process, especially in data-scarce scenarios. The results confirm that, in our case study, the OU process provides a more computationally efficient and accurate solution for synthetic human trajectory generation compared to TimeGAN.

As Industry 5.0 continues to advocate for human-centric, adaptive workplaces, this research contributes to the development of environments where robots and humans can collaborate seamlessly, with predictive systems playing a crucial role in enhancing safety, reducing downtime, and improving overall system performance.

ACKNOWLEDGMENT

It is supported by the Artificial Intelligence, Biomechatronics and Collaborative Robotics Research Group at the Top Research Center Mechatronics (TRCM), University of Agder (UiA), Norway.

REFERENCES

- [1] M. H. Zafar, E. F. Langås, and F. Sanfilippo, "Exploring the synergies between collaborative robotics, digital twins, augmentation, and Industry 5.0 for smart manufacturing: A state-of-the-art review," *Robot. Comput.-Integrated Manuf.*, vol. 89, Oct. 2024, Art. no. 102769.
- [2] F. Sanfilippo, M. H. Zafar, T. Wiley, and F. Zambetta, "From caged robots to high-fives in robotics: Exploring the paradigm shift from human-robot interaction to human-robot teaming in human-machine interfaces," *J. Manuf. Syst.*, vol. 78, pp. 1–25, Feb. 2025.
- [3] E. F. Langås, M. H. Zafar, and F. Sanfilippo, "Exploring the synergy of human-robot teaming, digital twins, and machine learning in Industry 5.0: A step towards sustainable manufacturing," *J. Intell. Manuf.*, vol. 37, no. 3, pp. 999–1022, Mar. 2026.
- [4] E. F. Langås, A. Aalerud, D. Hagen, and F. Sanfilippo, "Transformer-based human trajectory prediction in manufacturing settings," in *Proc. Eur. Robot. Forum*, 2026.
- [5] D. Helbing and P. Molnár, "Social force model for pedestrian dynamics," *Phys. Rev. E*, vol. 51, no. 5, pp. 4282–4286, 1995.
- [6] D. Helbing, I. Farkas, and T. Vicsek, "Simulating dynamical features of escape panic," *Nature*, vol. 407, no. 6803, pp. 487–490, Sep. 2000.
- [7] I. Karamouz, B. Skinner, and S. J. Guy, "Universal power law governing pedestrian interactions," *Phys. Rev. Lett.*, vol. 113, no. 23, Dec. 2014, Art. no. 238701.
- [8] H. Xue, D. Q. Huynh, and M. Reynolds, "SS-LSTM: A hierarchical LSTM model for pedestrian trajectory prediction," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 1186–1194.
- [9] J. Martinez, M. J. Black, and J. Romero, "On human motion prediction using recurrent neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4674–4683.
- [10] T. Petković, D. Puljiz, I. Marković, and B. Hein, "Human intention estimation based on hidden Markov model motion validation for safe flexible robotized warehouses," *Robot. Comput.-Integr. Manuf.*, vol. 57, pp. 182–196, Jun. 2019.
- [11] A. Löcklin, F. Dettinger, M. Artelt, N. Jazdi, and M. Weyrich, "Trajectory prediction of workers to improve AGV and AMR operation based on the manufacturing schedule," *Proc. CIRP*, vol. 107, pp. 283–288, Jan. 2022.
- [12] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras, "Human motion trajectory prediction: A survey," *Int. J. Robot. Res.*, vol. 39, no. 8, pp. 895–935, Jul. 2020.
- [13] M. N. Finean, L. Petrović, W. Merkt, I. Marković, and I. Havoutis, "Motion planning in dynamic environments using context-aware human trajectory prediction," *Robot. Auto. Syst.*, vol. 166, Aug. 2023, Art. no. 104450.
- [14] D. Mukherjee, K. Gupta, L. H. Chang, and H. Najjaran, "A survey of robot learning strategies for human-robot collaboration in industrial settings," *Robot. Comput.-Integr. Manuf.*, vol. 73, Feb. 2022, Art. no. 102231.
- [15] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2009.
- [16] Q. Wen, L. Sun, F. Yang, X. Song, J. Gao, X. Wang, and H. Xu, "Time series data augmentation for deep learning: A survey," in *Proc. 13th Int. Joint Conf. Artif. Intell.*, Aug. 2021, pp. 4653–4660.
- [17] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 2672–2680.
- [18] J. Yoon, D. Jarrett, and M. Van der Schaar, "Time-series generative adversarial networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 5508–5518.
- [19] J. C. F. Silva, M. C. Silva, V. J. P. Amorim, P. S. O. Lazaroni, and R. A. R. Oliveira, "Wearable sensors: Improving AI for walking activities through GAN-based data augmentation," in *Proc. 14th Brazilian Symp. Comput. Syst. Eng. (SBESC)*, Nov. 2024, pp. 1–6.
- [20] E. F. Langås, M. H. Zafar, S. O. Nyberg, and F. Sanfilippo, "Human trajectory simulation in industrial settings using the Ornstein-Uhlenbeck process and deep learning based classification," in *Proc. 10th Int. Conf. Autom., Robot. Appl. (ICARA)*, Feb. 2024, pp. 427–432.
- [21] G. E. Uhlenbeck and L. S. Ornstein, "On the theory of the Brownian motion," *Phys. Rev.*, vol. 36, no. 5, pp. 823–841, Sep. 1930.
- [22] F. Triggiano and M. Romito, "Gaussian processes based data augmentation and expected signature for time series classification," *IEEE Access*, vol. 12, pp. 80884–80895, 2024.
- [23] A. Aalerud, J. Dybedal, and D. Subedi, "Reshaping field of view and resolution with segmented reflectors: Bridging the gap between rotating and solid-state LiDARs," *Sensors*, vol. 20, no. 12, p. 3388, Jun. 2020.
- [24] V. Licaret, A. Aalerud, A. Belbachir, N. Belbachir, and M. Leordeanu, "Raphael: Realistic 3D data generation and unsupervised learning for enhanced human-robot detection in the real world," in *Proc. BMVC Scene Understanding Human Modeling Workshop*, 2025, Art. no. 10.
- [25] A. Aalerud and G. Hovland, "Dynamic augmented Kalman filtering for human motion tracking under occlusion using multiple 3D sensors," in *Proc. 15th IEEE Conf. Ind. Electron. Appl. (ICIEA)*, Nov. 2020, pp. 533–540.
- [26] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," *Phys. D, Nonlinear Phenomena*, vol. 404, Mar. 2020, Art. no. 132306.
- [27] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, no. 2, pp. 281–305, 2012.
- [28] *Robots and Robotic Devices—Collaborative Robots*, Standard ISO 15066:2016, 2016.
- [29] C. Elkan, "The foundations of cost-sensitive learning," in *Proc. Int. Joint Conf. Artif. Intell.*, 2001, pp. 973–978.
- [30] E. F. Langås, M. H. Zafar, and F. Sanfilippo, "Harnessing digital twins for human-robot teaming in Industry 5.0: Exploring the ethical and philosophical implications," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Dec. 2023, pp. 1788–1793.



EVEN FALKENBERG LANGÅS (Graduate Student Member, IEEE) received the B.Sc. and M.Sc. degrees in mechatronics from the University of Agder, Grimstad, Norway, in 2019 and 2021, respectively, where he is currently pursuing the Ph.D. degree in mechatronics. His background includes several years of experience in industrial software development, working with the design and implementation of systems for remote control, monitoring, and data acquisition in industrial environments. His research interests include machine learning, digital twins, and human-robot interaction and teaming in smart manufacturing.



tion systems for autonomous and intelligent robotic applications, including deployment in real-world industrial settings.

ATLE AALERUD received the M.Sc. and Ph.D. degrees in mechatronics from the University of Agder, Norway. He is currently a Senior Researcher with NORCE Research, focusing on robotics, computer vision, and real-time spatial sensing systems, emphasizing LiDAR-based and multimodal perception for industrial environments. His contributions include participation in several national and European research projects and patented innovations in LiDAR-based percep-



in various journals and conferences, within a variety of fields. He is also a member of the board of Norsk Matematikkråd (NMR), the Norwegian Advisory Council for Mathematics. He is the Chairperson of the Norwegian Statistics Association.

SVEIN OLAV NYBERG received the Ph.D. degree in mathematics (stochastic analysis) from the University of Oslo (UiO), Norway. He has moved from pure mathematics to applications of statistics. He is currently an Associate Professor with the Faculty of Engineering and Science, University of Agder (UiA), Grimstad, Norway. He takes part in many collaboration projects, offering his expertise in the fields of statistics and stochastics. He has authored and co-authored several technical papers



His research interests include robotics, machine vision, human-robot teaming, and swarm optimization. He is an IEEE Student Section Representative of the Norway Section. He is also a reviewer of several international conferences and journals.

MUHAMMAD HAMZA ZAFAR (Graduate Student Member, IEEE) received the bachelor's degree in electronics engineering and the master's degree in electrical engineering from the Capital University of Science and Technology, Islamabad, Pakistan, in 2015 and 2022, respectively. He is currently pursuing the Ph.D. degree in robotics and AI with the Department of Engineering Sciences, University of Agder, Grimstad, Norway.



Group (iTron). He is the Co-Founder and the Chair of the Norwegian Society of Mechatronics. His background includes several years of industrial experience in offshore and marine control systems and systems engineering. His research interests include autonomous robotics, intelligent digital twins, distributed control systems, and AI-assisted engineering, with emphasis on safety-critical offshore and maritime applications. His contributions include participation in national research projects funded by the Research Council of Norway and patented innovations in autonomous load-handling systems. He is a member of the Advisory Board of the Global Fluid Power Society. He is an Associate Editor for *MIC Journal*.

DANIEL HAGEN (Member, IEEE) received the joint M.Sc. degree in mechatronics from the University of Agder (UiA), Norway, and FH Vorarlberg, Austria, in 2014, and the Ph.D. degree in engineering science with specialization in mechatronics from UiA, in 2020. Currently, he is an Associate Professor with the Department of Engineering Sciences, UiA, where he is also the Director of the Norwegian Motion Laboratory and the Head of the Intelligent Mechatronics research



various national projects from the Research Council of Norway (RCN). His research focuses on human-robot teaming (HRT). He is a reviewer for several international conferences and journals. He is the Former Chair of the IEEE Norway Section. He is the Chair of the IEEE Robotics and Automation, Control Systems, and Intelligent Transportation Systems Joint Chapter. He is the Chair of the Norway Section Life Members Affinity Group. He is a member of the IEEE Region 8 Chapter Coordination Committee, the Conference Coordination Committee, the IEEE Public Visibility Committee, the IEEE R8 Awards and Recognitions Committee, and the Professional and Educational Activities Committee. He is also the Former Treasurer of the Norsk Forening for Kunstig Intelligens (NAIS), the Norwegian Association for Artificial Intelligence.

FILIPPO SANFILIPPO (Senior Member, IEEE) received the Ph.D. degree in engineering cybernetics from the Norwegian University of Science and Technology (NTNU), Norway. He is currently a Professor with the Faculty of Engineering and Science, University of Agder (UiA), Grimstad, Norway. He has authored and co-authored several technical papers in various journals and conferences. He carries a vast experience in participating in European research programs and

...