This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No. 899739

CROWD DNA

H2020 EU Fet-Open Project

Technologies for computer-assisted crowd management

www.crowddna.eu

Call: H2020-FETOPEN-2018-2019-2020-01

Type of action: RIA

Grant agreement: 899739

WP N°4:	**Crowd management technologies**
Deliverable Nº4.2:	**Crowd management solutions**
Task lead:	**UCL**
WP lead:	**CDI**
Version Nº:	**1.0**
Date:	**29/08/2024**

Document information	
Deliverable N° and title:	D4.2 – Crowd management solutions
Version N°:	1.0
Task lead:	UCL
WP lead:	CDI
Author(s):	Feixiang He (UCL), Jialin Zhu (UCL), He Wang (UCL) , Alex Elms (CDI), Paul Townsend (CDI)
Reviewers:	Paul Townsend (CDI), Julien Pettré (Inria), Solenne Fortun (Inria)
Submission date:	29/08/2024
Due date:	31/08/2024
Type:	R
Dissemination level:	PU

Document history

Date	Version	Author(s)	Comments
28/06/2024	0.1	Solenne Fortun	Document creation
09/08/2024	0.2	Feixiang He, Jialin Zhu, He Wang	First version
12/08/2024	0.2	Julien Pettré	Review
13/08/2024	0.2	Solenne Fortun	Review
16/08/2024	0.3	Feixiang He, Jialin Zhu, He Wang	Revision after review
23/08/2024	0.4	Paul Townsend	Review
27/08/2024	0.5	He Wang	Revision after review
29/08/2024	1.0	Solenne Fortun	Final layout

Table of Contents

Table of Figures

Table of Figures

Acronyms and Abbreviation

CDI	Crowd Dynamics International Limited
EC	European Commission
EMT	Executive Management Team
FZJ	Forschungszentrum Julich Gmbh
GA	Grant Agreement
INRIA	Institut National de Recherche en Informatique et Automatique
KPIs	Key Performance Indicators
PO	Project Officer
UL	University of Leeds
ULM	Universität Ulm
URJC	Universidad Rey Juan Carlos
WP	Work-Package

Executive Summary

The overall objective of WP4 is *to explore usage of the main technologies developed along the project CrowdDNA for crowd analysis, and based on which to develop management techniques that explore the best usage of this information".* This deliverable, D4.2, describes the development of modelling techniques and their uses for new crowd management solutions.

In previous WPs (WP1-3), data and models have been developed to analyse individual and crowd motions as a whole in both moderate and high density. In WP2, the individual motion analysis includes modelling intensive physical interactions in high-density crowds, on people's reactive motions under unexpected physical disturbances, such as pushes, and how these reactive motions can propagate through crowds. This kind of propagation, albeit starting from individuals, can lead to amplified turbulence-like motions in high-density crowds and ultimately crushes. In WP3, video-based tools have also been developed to analyse and predict how a small crowd as a whole and how individuals in it will move together with others. These tools combined provide the ability for online and offline simulation and prediction to make a trigger for the authority to decide whether urgent intervention needs to be conducted, e.g. overly crowded, or potential crushes, etc.

While tools in WP2-3 provide good low-level prediction (e.g. a small area), D4.2 focuses on testing and potential generation of high-level crowd management strategies. It aims to progress techniques that can be used as tools for the authority to monitor, predict and test their management strategies in a big space with real-world settings.

1. Introduction

1.1. Background

In CrowdDNA, we aim to predict and analyze human movement at multiple levels, each addressing specific aspects of human and crowd dynamics. The multiple level modeling effort is demonstrated in WP2, WP3 and WP4, each focusing on a different level. In WP2, we forecast individual trajectories and predict full-body motions at a microscopic level by incorporating explicit physics models with and without deep learning techniques. This covers the fundamental individual body physics and how physical interactions between individuals can affect people and propagate in a crowd. In WP3, the effort is to link individual motions in WP2 to macroscopic crowd dynamics. This has led to two video-based analysis tools that can take crowd videos at relatively close distance where some body motions can still be seen, or at a relatively far distance where individuals are a small number of pixels, and predict short-horizon future.

In WP4, the aim is to go further up in scale and model crowd activities in large public spaces. This level of modeling is aiming to provide management tools for the event organizers and authorities. Different from previous tools in WP2 (individual level) and WP3 (local area), the data needed for WP4 is a large area where only partial observations can be obtained, e.g. sensors distributed in a space but can only cover a certain portion of the whole space, which is a common setting in many public spaces.

Specifically, we choose IoT data, as detailed in WP4.1, for our experiments due to its ability to reveal density variations across different regions. Although video data and trajectory data are useful for analyzing human movements and interactions, they become infeasible for predicting crowd density changes in large public spaces. Trajectory data may not provide sufficient insight into the collective behavior of a dense crowd, and acquiring accurate trajectories is particularly challenging in high-density scenarios. Video data, on the other hand, is typically constrained by its field of view and resolution, limiting its ability to capture the full scope of a large crowd's movement. Predicting crowd density changes for effective crowd management requires a macroscopic approach that can account for the collective behavior of thousands of individuals across the entire scope of the space, and IoT data is well-suited for this purpose. The specific data we use is from INNOCESS, where details can be found in WP4.1.

1.2. Motivation and Challenges

The goal of this work is to find suitable methods that enhance crowd management in a large space, with scarce sensing, with potential high-density crowds. This is commonly seen in public spaces such as music festivals, sporting events, religious gatherings, public transportation hubs and emergency evacuations [1]. Effective crowd management strategies can prevent accidents, reduce stress, and improve the overall experience for participants. By understanding and predicting crowd dynamics, organizers can design safer venues, optimize evacuation routes, and implement timely interventions to manage crowd flow and prevent potential disasters such as stampedes or crowd crushes. Recently, the increasing frequency and scale of large public events show the need for advanced crowd management techniques that can handle complex and dynamic environments.

However, managing crowds in a large space presents significant challenges. One of the primary difficulties is the unpredictable nature of human behavior and the complex interactions within large groups. On an individual level, physical contact is inevitable, leading to reactive rather than controlled movements. These reactive movements can propagate through the crowd, potentially resulting in dangerous situations. For instance, a push from the back of a crowd can travel forward, causing a chain reaction that might result in a crowd collapse. Furthermore, in densely crowed environments, individuals have limited space to recover balance, making it difficult to stabilize themselves without affecting others. This lack of space can increase the risk of accidents and injuries. Another significant challenge is the scarcity and quality of data. Real-world high-density scenarios are complex and dynamic, making it difficult to capture accurate and comprehensive data on crowd movements and interactions. Ethical and privacy concerns often limit the extent of data collection in public spaces, further impeding the development of precise predictive models and limiting the ability to validate and refine crowd management strategies.

Current crowd-management approaches are inadequate because they often rely on overly simplistic models. These models overlook the complex interactions and sudden changes that can occur within a crowd, failing to capture the dynamic and unpredictable nature of human behavior in high-density environments [2]. These models generally assume simple movement and rational decision-making, overlooking the complex interactions and sudden changes that can occur within a crowd. Additionally, traditional methods often utilize limited and static data sources, which fail to provide real-time insights necessary for responsive management. Most existing research cannot adequately address the sensing sparsity in big spaces, where only isolated areas can be observed, and the whole crowd state is often not directly observable. The lack of integration with advanced technologies like AI and machine learning further reduces the effectiveness of these approaches, making them inadequate for preventing incidents and ensuring safety in increasingly complex and crowded scenarios.

To address these challenges, innovative approaches are needed to enhance data collection and model development for high-density crowd management. Advances in sensor technologies, such as wearable devices and high-resolution video analysis, can provide richer datasets that capture the minor differences of crowd behavior. Incorporating machine learning techniques can help process and analyze complex data, enabling more comprehensive and accurate models. By leveraging these advancements, crowd management can evolve to effectively handle the demands of high-density scenarios, ultimately creating safer and more enjoyable experiences for all participants.

In this work, we primarily focus on crowd management in high-density scenarios in a big space. Our tool is capable of predicting density changes in observed regions and could act as an early warning system for potential dangers when crowd density in a region exceeds a certain threshold. Additionally, our tool can suggest effective evacuation strategies, such as directing crowd flow to safer areas, when potential dangers are detected, via predicting how the density of the whole space is likely to change if certain decision are made, e.g. evacuate people from one area to another.

2. Proposed Solution

2.1. Settings and Preliminaries

Our primary focus is on modelling methods that can assist crowd management in large-scale high-density public spaces. Specifically as a case study, our setting and data come from the world-renowned Hellfest, a rock music festival, shown in **Figure 1**. By deploying sensors at key locations, the time-varying crowd density data from these sensors is obtained. Due to that the real-world data is limited in size and variety, additionally, we simulate the Hellfest scenario using crowd simulation software to generate synthetic data.

We point out that significant challenges in data scarcity still exist even with synthetic data. In both real and synthetic data, the obtained data lacks global information due to that the sensor network in Hellfest only covers isolated areas. This is because even in simulation, although we simulate agents in the whole space, we assume that we can only get readings from the sensors (the green dots in **Figure 1**). Therefore, we need a model that can deal with local information per sensor and learn the relationships between different locations from data.

A Graph Neural Network (GNN) fits this requirement well. We treat the sensor network as a graph, where sensors are graph nodes, but their connectivity is unknown. This is because people can randomly walk into and leave the sensor areas and the way they walk through different sensing areas can be affected by many factors. This means when people leave one sensor area, they might not appear in the closet next sensor but suddenly appear in a sensor that is far from the current one later. As a graph, this means that there is no good way to predefine the connectivity of sensors in the graph, as a proximity measure of different sensors. We propose to infer the connectivity from data. However, this approach introduces a new challenge: since the festival is held in an open area, the relationships between nodes are not solely distance-dependent but vary in different situations (entry, in progress, resting, exit, etc.). In this context, it is evident that a model capable of dynamically creating and inferring new graph structures based on previous states would be more effective.

Figure 1. Map of the Hellfest venue.

The google earth map of Hellfest venue is shown in **Figure 1**, where unrelated areas are masked out by grey. Green dots are sensors that have a sensing capacity of a certain radius. The bottom right corner is the main stage where the sensor density is relatively high. In other parts, the sensors are sparse. The top two dots connect the space to another big open space. The left most three sensors are the main entrance at the bottom and a camping site on the left, respectively. The two sensors in the middle are installed in a bottleneck area.

2.2. Model Overview

As mentioned before, the crowd dynamics prediction of Hellfest data is based on fixed positions and time sequences. So, we design a model based on TrendGCN [3], that can learn both spatial and temporal relationships for our specific situation. The model consists of a generator for dynamic adaptive graph generation and two discriminators. The generator is responsible for capturing dynamic spatial dependencies, and two discriminators are used to evaluate and eliminate the trend-level and dependency-level discrepancies. The overall architecture is shown in **Figure 2**.

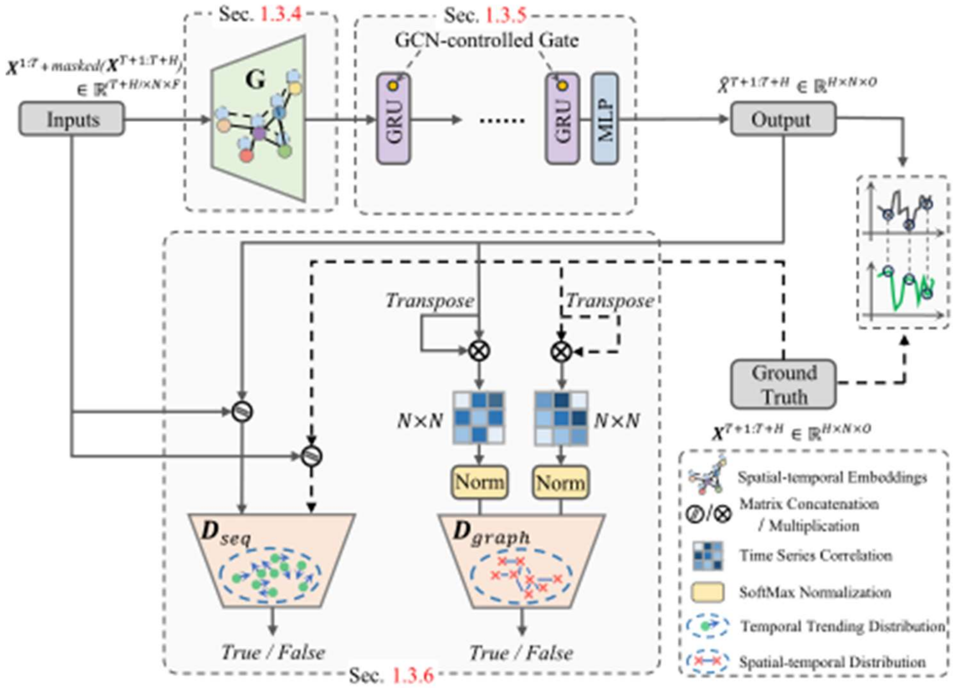

Figure 2. Overview architecture of Model.

2.2.1. Definition

The crowd dynamics is defined as $X^{1:T} = \{X^1, X^2, \ldots, X^T\} \in R^{T\times N\times F}$. The time sequence length is T, the node number is N, and the density feature size is F. Each X_i^t represents the crowd density value of nodes i at time step t, i.e. a sensor at a time. Our model is utilized to learn the prediction of $X^{\{T+1:T+H\}}$ given $X^{\{1:T\}} + \text{masked}\left(X^{(T+H)}\right)$as input. The masking function eliminates some of the data from $T + 1: T + H$ by a mask. We use $T = 4$ and $H = 4$ during practical experiments.

2.2.2. Dynamic Adaptive Graph Generation

The vanilla version of the Graph Neural Network calculates the embedding properties of each node based on a pre-given adjacent matrix. However, it is limited to capturing static graphs and cannot represent the changing spatial dependencies between nodes at different time steps. To enable the graph neural network to dynamically adjust the value of the adjacency matrix according to time, there are two types of embedding, spatial embedding and temporal embedding to denote the unique representations of each node and each time step.

Then a unified approach that efficiently combines spatial (node-wise) and temporal (time-wise) embedding with a gate module is utilized to construct dynamically evolving graphs. In detail, we add the spatial and temporal embedding of each node and perform Layer Normalization and Dropout operation on the result. The output of Layer Normalization and Dropout operation is the integrated embedding for dynamic graph generation. The adjacency relationship between two nodes is then connected through the inner product of integrated embedding. This approach enables not only homogeneous interactions within the spatial and temporal domains separately but also facilitates direct interactions between node embedding and different time steps. The constructed graph can simultaneously represent spatial, temporal, and spatial-temporal interactions, offering a stronger representational capability compared to a static adaptive graph that only focuses on spatial interactions. Finally, we use a 1st-order Chebyshev polynomial expansion to approximate graph convolution. This approximation utilizes parameters specific to the combinations of spatial and temporal embedding.

2.2.3. Dynamic Graph Convolutional GRU

In accordance with previous studies [4, 5], the proposed Dynamic Adaptive Graph Generation module is integrated into Gated Recurrent Units (GRU) [6] by substituting the MLP layers within the GRU. Subsequently, multiple GRU layers are stacked and followed by a linear transformation (MLP) to project the T_{th} output of GRU for generating H steps ahead predictions in a sequence-to-sequence format. This method notably reduces both time costs and error accumulation.

2.2.4. Adversarial Dynamic Trend Alignment

Graph-based flow prediction often uses MAE (Mean absolute error) as the training loss function. However, we find that using only MAE is not enough as it treats each predicted result individually and cannot take the trends for global constraints. For example, two identical MAE values may correspond to different results that one is stable and the other is oscillating.

To solve this problem and better learn the changes in crowd dynamics trends, we introduce two discriminators with adversarial training to account for global properties, such as trends and inherent statistical correlations. These discriminators systematically assess discrepancies at both the trend and dependency levels, thereby enhancing robustness. The sequence discriminator specifically targets the trends within individual time series, while the graph discriminator focuses on the correlations among multivariate time series. L1 loss is used as the training objective and is jointly optimized with the adversarial training loss for the generator to facilitate multi-step predictions.

2.2.5. Control of Prediction

So far, the model only supports prediction. However, to better achieve crowd management objectives, it is essential to be able to input a control signal that can influence crowd dynamics. This is particularly applicable for applications such as path planning in emergency evacuation scenarios. Research on control signals based on Graph Neural Networks is extremely rare because modifying the adjacency matrix weights between nodes to adapt different behaviors through simple signals is very challenging. Therefore, a method similar to those used in Computer Vision for inpainting tasks has been adopted to achieve a certain degree of control over the prediction results.

We modified the input data dimensions of the model, changing from predicting $X^{T+1:T+H}$ using $X^{1:T}$ to predicting $X^{T+1:T+H}$ using $X^{1:T+H}$. In some of the dimensions in the input data $X^{T+1:T+H}$, some nodes (ranging from one-third to one-half) are randomly masked out. With this setup, our model can predict the complete $X^{T+1:T+H}$ based on $X^{1:T}$ and the given incomplete $X^{T+1:T+H}$. After training the model, the values of the unmasked nodes in $X^{T+1:T+H}$ can be manually adjusted as additional control signals to control the output results and accomplish the simulation tasks.

3. Results

3.1. Datasets

Data for our problem is scarce, especially in extremely high-density scenarios. Collecting real-world data can be expensive and may raise ethical and privacy concerns. Therefore, we choose synthetic data for conducting experiments and developing models, as it is both risk-free and cost-effective. In this case, we have chosen Hellfest as our experimental scenario because it is one of Europe's largest and most densely attended music festivals, featuring a complex layout and intricate human behaviors, which provide a realistic and challenging environment for testing and refining crowd management strategies. As depicted in **Figure 3**, the left shows the original selected scene from Hellfest, while the right illustrates its semantics, including the unwalkable areas (gray regions) and five entrances/exits (blue circles).

To create the synthetic dataset, we generate six 1-hour crowd simulations using ORCA [7] in Menge[1]. Each simulation features a random number of individuals, ranging from 2750 to 3850, entering the scene via an entrance within the first 45 minutes. The simulations include two distinct behaviors observed in the real scene: 1) entering the scene and immediately leaving through an exit; 2) going to the main stage, waiting for a while, and then proceeding to an exit. By deploying sensors within the scene, shown in **Figure 4** (Left), we can collect the number of individuals entering the sensor-captured areas within a 15-second interval, which we refer to as density in our experiments. Note that the perceived radius for the sensors in front of the main stage is 20 meters, while for the rest of the sensors, it is 10 meters. After preprocessing, we feed the density data into the predictive network. Moreover, we generate an evacuation dataset consisting of 2074 individuals, most of them start standing in front of the main stage, then randomly choose an exit to evacuate. We calculate then density data as the same way introduced above.

We also create experiments on a real-word dataset which is captured by Inocess[2]. The sensor distribution in the real data differs from that in the synthetic data, as shown in **Figure 4** (right). The perceived radius of all sensors is 20 meters. Moreover, the density data is provided and can be directly fed into the network. Note that the number of sensors is smaller in this dataset due to that Inocess combined some sensor readings. The sensor network itself is still the same as in **Figure 4** left.

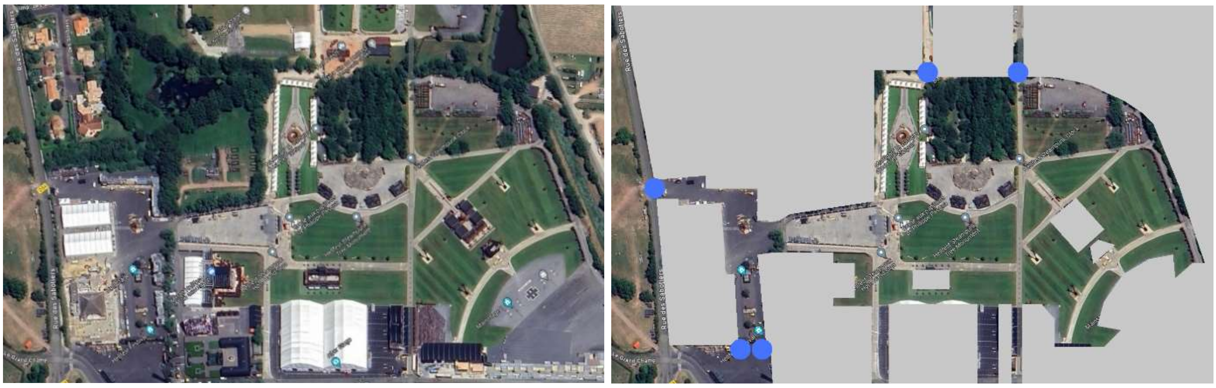

Figure 3. Left: original scene of Hellfest; Right: unwalkable areas and entrance.

1 https://gamma.cs.unc.edu/Menge/

2 https://www.inocess.com/

Figure 4. Left: sensor distribution in synthetic data; Right: sensor distribution in real data.

<u>Challenges in partial observation of crowds.</u> To comprehensively understand the collected dataset and to develop and fine-tune predictive models, we randomly selected one simulated dataset to analyze its statistical properties. Specifically, for each time interval, we counted the number of people observed and those not observed in the previous interval. As illustrated in **Figure 5**, the left image displays the ratio of unobserved people to observed people, while the right image presents the ratio of unobserved people to the total number of people in the simulation.

The plots show first show that we mimic real-world scenarios where the sensor network does not observe the whole crowds, so the synthetic data has this partial observation nature. As time passes by, the activities we simulated are people gathering in front of the main stage so that most of them are observed. This is qualitatively the same as our observations of Hellfest. Note that it is challenging to plot similar statistics for the Inocess data as we do not know how many people are unobserved. Finally, this analysis illustrates that predicting crowd densities is a challenging task due to the significant number of individuals who are not observed in the previous interval.

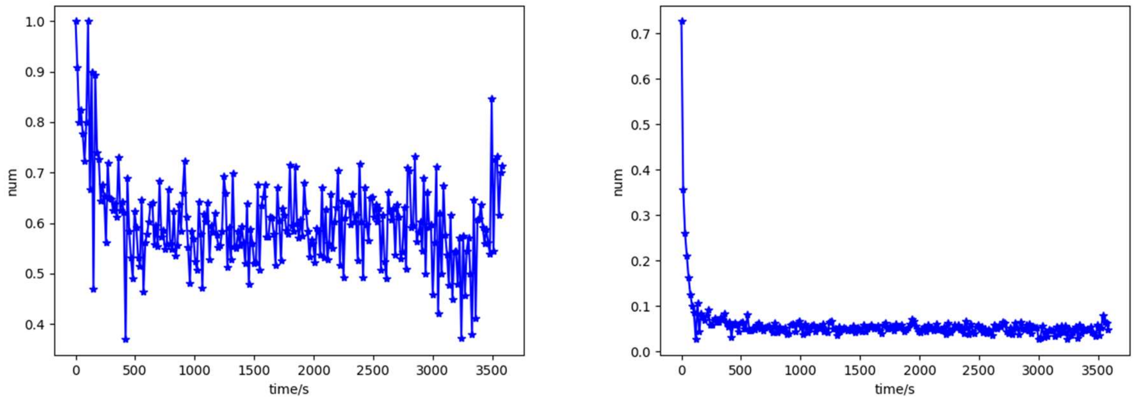

Figure 5. Left: the ratio of unobserved people to observed people; Right: the ratio of unobserved people to the total number of people.

3.2. Experiments

For all experiments, we chose to use 8-time steps as input (the data of first 4-time steps are complete, and data of the last 4-time steps are randomly masked for some nodes) to predict the node density information missing from the last 4-time steps in the input data. The Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) are used for evaluating the performance of the model.

3.2.1. Synthetic Crowd Data Experiment

The specific settings of this experiment have been described in the previous subsection. We use 4 days of data for training, 1 day of data for validation, and 1 day of data for testing in the 6-day simulation. The performance of our model on the test data is shown in **Table** 1.

Table 1. Results of synthetic crowd data experiment.

Time step	MAE	RMSE
1	0.56	1.25
2	0.63	1.37
3	0.64	1.43
4	0.66	1.45
Average	0.62	1.38

It can be observed from the results that our predictions on the test set are highly accurate, with an average error of only 0.5-0.6 people on each node. Since our time series data includes many nodes with values of 0, we were unable to calculate the Mean Absolute Percentage Error (MAPE). However, to demonstrate the quality of our results, we calculated the average value for each node- 5.052, providing an estimate of the percentage-based error- 12.27%.

To validate the effectiveness of our model in data distributions outside of the training data, we simulated an evacuation scenario specifically that the crowd would rush to the exit more aggressively compared to normal exit behavior. The performance is shown in **Table 2**.

Table 2. Results of evacuation test with synthetic data trained model.

Time step	MAE	RMSE
1	10.09	20.04
2	9.88	19.6
3	9.67	19.36
4	9.51	19.18
Average	9.79	19.55

Our model's performance on evacuation scenarios is worse than before, with errors increasing to around 9-10 compared to the training and test datasets that were within the same data distribution **Table 1**.

This is understandable as we intentionally exclude the evacuation data from the training, so that the distribution shift between our training data (normal entering and exiting) and testing data (aggressive evacuation) is huge. However, we argue that our model can still predict the overall trend under evacuation, which is significantly different its prediction on normal behaviors. Since we do not have data from Hellfest on evacuation, we cannot test our model prediction on real world evacuation scenario. However, we argue if the data is available to the organizers, they can incorporate it in training the model.

3.2.2. Real-world Crowd Data Experiment

The biggest difference between Hellfest data and our simulation data is that Inocess collects data for a longer period. Although there are only 4 days of data, these data cover 24 hours a day, unlike our simulation which only has 1 hour of data. Therefore, for the division of training, validation, and test data, we take the first 80% as training data, 10% as validation data, and the last 10% as test data in chronological order. The performance of our model on the test data is shown in **Table 3**. We also test this trained model with the Evacuation scenario. The performance is shown in Erreur ! Source du renvoi introuvable.

Table 3. Results of real-world crowd data experiment.

Time step	MAE	RMSE
1	2.84	6.59
2	3.59	8.09
3	4.29	9.58
4	4.44	10.49
Average	3.79	8.81

The results show that our predictions on the real-world data set still illustrates very strong performance, with an average error of 2-4 people per node. Although this error is higher compared with the experiments with simulated data, considering that the real-world data captures a greater number of crowd density measurements at each node, the result is actually quite competitive. Similarly, because the time series data includes many nodes with values of 0, we were unable to calculate the MAPE. However, to demonstrate the quality of our results, we calculated the average value for each node, 120.05, providing an estimate of the percentage-based error, 3.16%. Because the amount of real-world data far exceeds our simulated data, our model achieves better results.

Table 4. Results of evacuation test with real-world data trained model.

Time step	MAE	RMSE
1	88.64	133.91
2	88.77	136.79
3	90.29	142.63
4	92.04	148.37
Average	89.93	140.54

When we tested the evacuation scenario using the model trained on real-world data, we observed a significant drop in performance, with the average error reaching around 88-92. This is primarily because the Hellfest data does not contain crowd dynamics similar to those in an evacuation scenario. Additionally, the simulated evacuation scenario involves a completely different crowd size compared to the actual crowd numbers in the Hellfest data. But we argue this is not the flaw of our model but simply lack of training data on different behaviors.

After evaluating our model on both real and synthetic data, we notice a difficulty in model generalization when the data shift is large between training and testing data. This is a known long-standing issue in the field. The key issue is the lack of data. Often, the organizers are not at the liberty of sharing data with the model developers, and they usually do not have data of extreme behaviors either (e.g. aggressive evacuation). Given that we will share our model and give specific instructions on model training and application, the former difficulty can be addressed in house by the organizers. The latter could be mitigated by using simulation data to augment real data.

3.2.3. Auto-regressive Experiments

We also conducted conditioned autoregressive prediction, to demonstrate that we can use control signals to enable the model to simulate crowd dynamics to unseen scenarios to some extent. Using the model trained on real-world data, we took the data of first 8-time steps with masked information from test set as input. After obtaining the predicted complete data for the subsequent 4-time steps, we manually reduced the values for the nodes representing the food area and main stage area, and masked the values for the remaining nodes. This manual reduction on the density is essentially a control signal aiming to reduce the density of these areas. This modified set data of 4-time steps, along with the predicted data, was then concatenated as the new input for further time step predictions. By repeating this process, we can show how the crowd density at other nodes changes when the density at the food area and main stage nodes is reduced in **Table 5**.

Table 5. Results of Auto-regressive experiments.

Time	Entrance crowd density	Camping area crowd density	Cathedral crowd density	War zone crowd density	Food area crowd density	Main stage crowd density
0 min	124	58	141	49	198	448
4 min	120.0748	64.1197	146.9948	52.6297	186.9011	408.7964
8 min	121.3542	66.2470	150.2825	52.8685	174.4680	344.1492
12 min	125.6571	69.0952	154.9834	53.9146	163.1991	289.0938
16 min	131.7673	70.8468	158.6927	54.3397	152.4008	255.0702
20 min	139.4122	71.2164	160.4811	54.0742	140.5723	232.5807

Testing of 'what-if' situations. Our auto-regressive test results demonstrate that if the users aim to reduce the density of some areas, our model can predict what is likely to happen to other areas in terms of their densities. In this setup, manually reducing the crowd density at the food area and main stage area nodes corresponds to making people to go to other areas. Our results also show that in such cases, the crowd density at the entrance, camping area, and cathedral area increases, as expected in real-world situations. However, we acknowledge that the prediction is merely qualitative and projections of what could happen. We show the visualization of this experimental result in **Figure 6**. Different colors are used in the figure to represent the change of crowd density. The detailed information of color and crowd change is shown in the color bar on the right. According to the degree of increase in the number of people, the color will move towards dark purple, while the degree of decrease in the number of people will move towards light yellow.

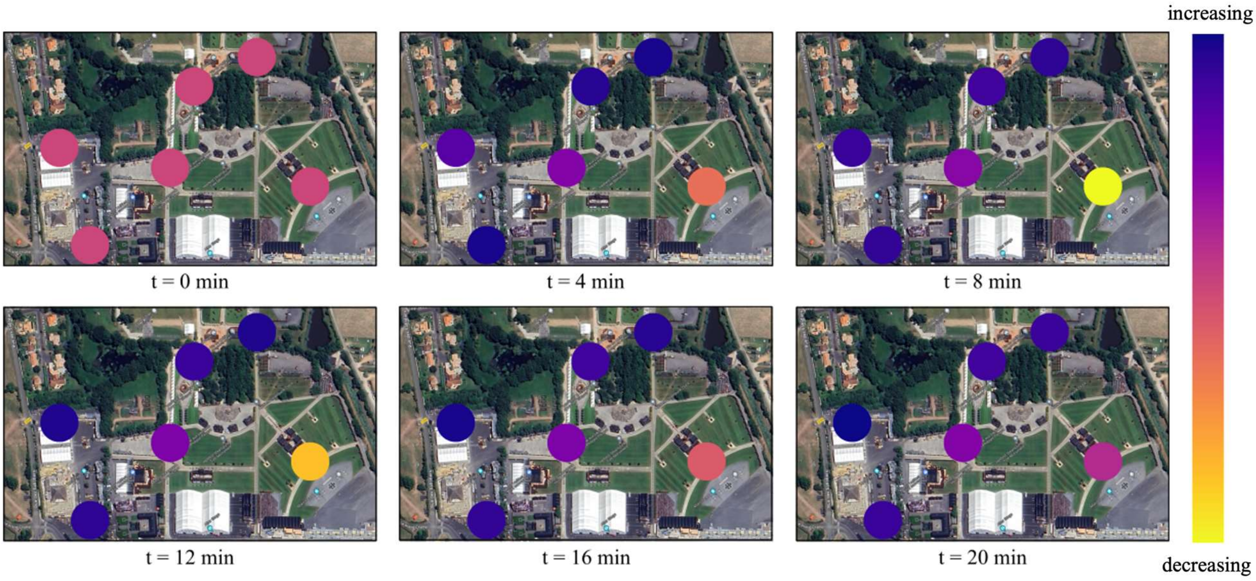

Figure 6. Visualization of crowd density change in auto-regressive experiment.

4. Crowd Management Applications

Whilst the above focuses on methods and innovation to assist crowd management strategies, the potential use of these in the industry should be considered.

Crowd management itself can be defined as an integrated concept for organizing, designing and operating a gathering of people in a limited space. In practice, this means that the design of sites will be analysed and planned for large crowds in terms of capacity, crowd movement through ingress, circulation and egress. During the event, operations enact the crowd management plan, often requiring dynamic decisions based on CCTV footage or reports from staff on the ground. Some events and venues use technology to monitor crowds, but there is little to actively analyse crowd management strategies and actively help to predict the impact of decisions leading to greater situational awareness and better decision making.

The techniques developed in this deliverable can be utilised into two main analysis/decision making processes where they can be applied to monitor crowds, develop crowd management strategies or implement evacuation or rerouting. These are as follows:

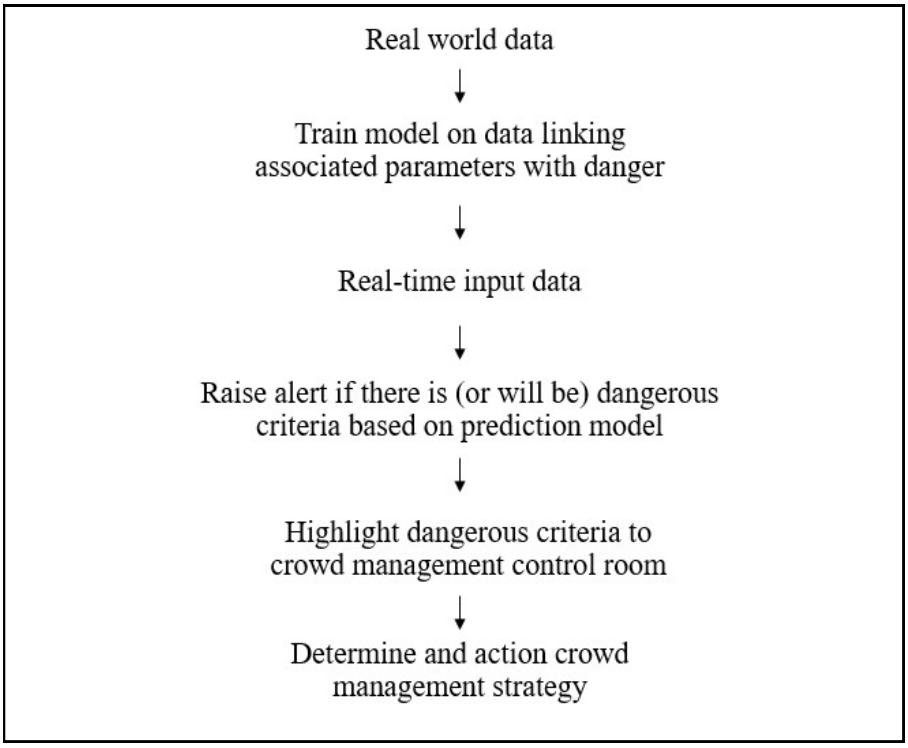

Figure 7. Process 1: Predictive alerts

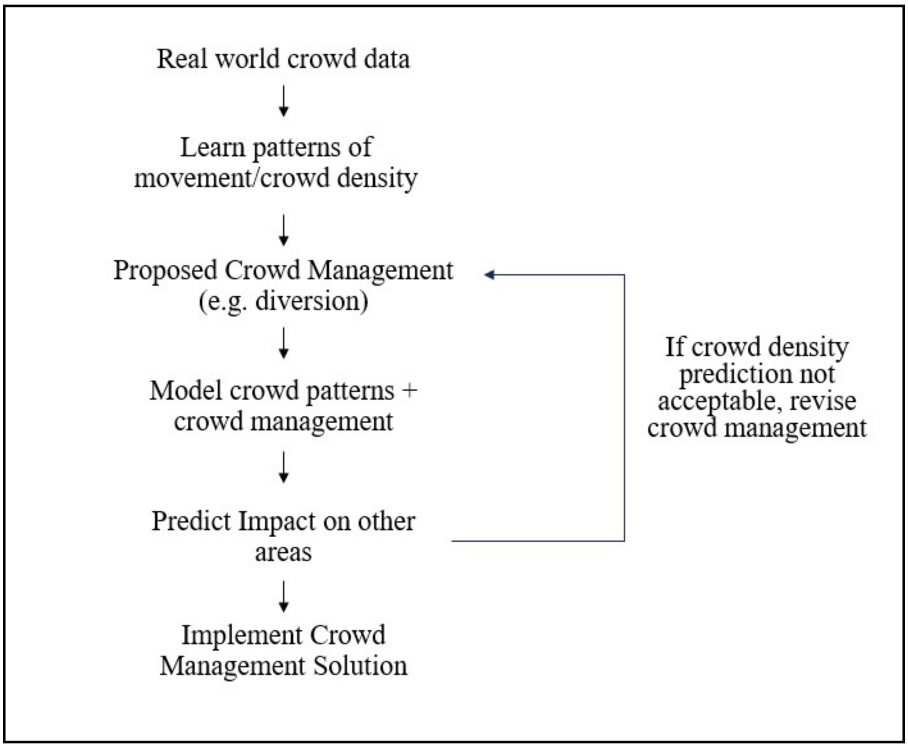

Figure 8. Process 2: Iterative Crowd Management Strategy Development

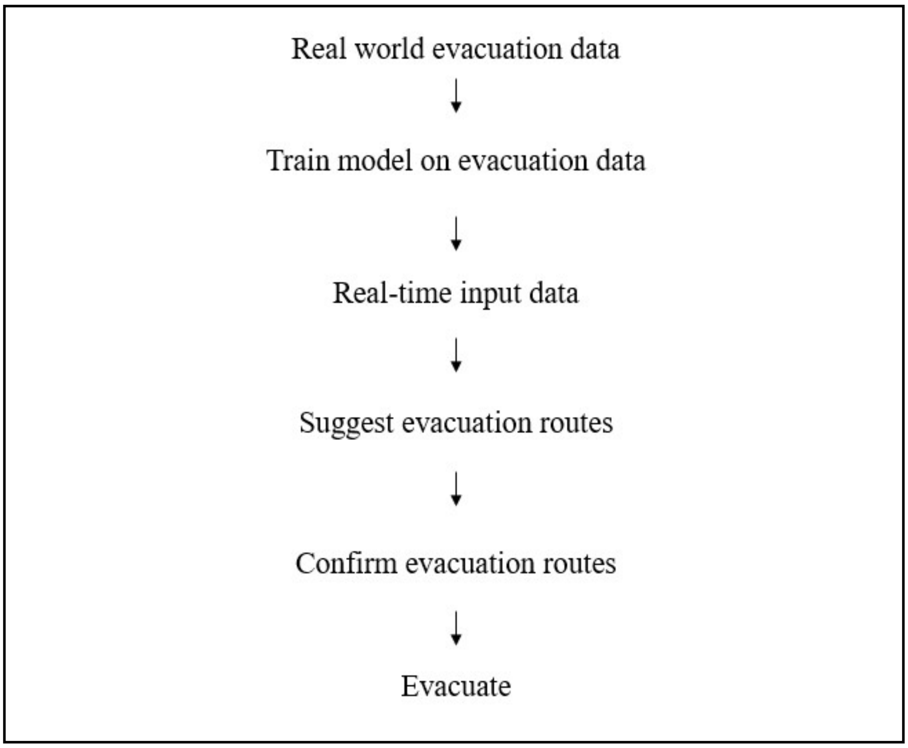

Figure 9. Process 3: Generative Rerouting

The processes are very high level and purposefully generic as the models could be applied to a number of different crowd management situations and strategies such as prediction of density, area fill rates, routing, normal egress and evacuation. It is important to note that these processes overall are unlikely to be fully automated, because the safety of people is paramount, a human decision maker should be ‘in the loop’ to verify any generated strategies or test results.

Process 1 (see **Figure 7**) would ideally suit a real-time use where alerts forewarn of potential danger. The main benefit is that a human cannot realise the model outcomes by logic or experience alone, it would provide new information that feeds into a decision process.

Process 2 (see **Figure 8**) is best used during planning stages and for analysing data post-event to learn lessons. The iterative nature is less suitable for real-time use. It would allow the models to be used for testing and optimising crowd management plans.

Process 3 (see **Figure 9**) can be used for normal crowd routing or evacuation, where the impact of route planning could be tested or an optimum evacuation or rerouting strategy can be generated based on the learning model. This can be used at all stages of analysis and crowd management.

Deploying Process 2 and Process 3 as part of crowd management planning would allow for a feedback loop for regular events year on year or day by day as the event goes on. It would also give a comparative and better understanding of how to manage crowds for fixed venues (e.g. stadia, transport hubs) that see different crowd movements and densities for different events or times of day and year. In this way, they can be used as both planning, during operation and for post-event assessments, assuming that the technology can develop further into a usable tool (out of scope of this project).

5. Conclusion

This deliverable has described the CrowdDNA crowd management models that are aimed towards being part of a decision-making process to determine crowd management strategies, before, during or after events. This is specifically for managing crowds in high-density and wide area scenarios. Our pipeline not only serves as for early warning of potential dangers but also provides a way of predicting the likely crowd movements if certain intervention is conducted. We have demonstrated the effectiveness of our pipeline on both synthetic and real-world data.

In future, we plan to share the code. Additionally, we aim to refine our methodology by incorporating invisible nodes within the graph model, allowing us to capture unobserved information within the scene. We will also explore different methods of controlling signals, offering more precise and adaptive crowd management strategies. These enhancements will contribute to a more robust and flexible framework for predicting and managing crowds in large public spaces.

References

[1] O'Toole, William. Crowd Management: Risk, security and health. Goodfellow Publishers Ltd, 2019.

[2] Krishnakumari, Panchamy, et al. "Crowd Safety Manager: Towards Data-Driven Active Decision Support for Planning and Control of Crowd Events." arXiv preprint arXiv:2308.00076 (2023).

[3] Jiang, J., Wu, B., Chen, L., Zhang, K. and Kim, S., 2023, October. Enhancing the robustness via adversarial learning and joint spatial-temporal embeddings in traffic forecasting. In Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (pp. 987-996).

[4] Bai, L., Yao, L., Li, C., Wang, X. and Wang, C., 2020. Adaptive graph convolutional recurrent network for traffic forecasting. Advances in neural information processing systems, 33, pp.17804-17815.

[5] Yu, H., Li, T., Yu, W., Li, J., Huang, Y., Wang, L. and Liu, A., 2022. Regularized graph structure learning with semantic knowledge for multi-variates time-series forecasting. arXiv preprint arXiv:2210.06126.

[6] Chung, J., Gulcehre, C., Cho, K. and Bengio, Y., 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555.

[7] Van den Berg, Jur, Ming Lin, and Dinesh Manocha. "Reciprocal velocity obstacles for real-time multi-agent navigation." 2008 IEEE international conference on robotics and automation. Ieee, 2008.