proceedings

MDPI

Proceeding Paper

Dynamic Evacuation Shelter Allocation in Response to Human Mobility: A Case Study of Taipei City †

Chang-Hung Shih [1,*], Cheng-Yun Wu [2], Shu-Ping Tseng [2], Yi-Lin Huang [2], Rong-Pu Jhuang [2], Yi-Chung Chen [1], Tien-Yi Yang [2] and Wei-Ting Chen [2]

[1] Department of Computer Science and Engineering, National Chung Hsing University, Taichung 402202, Taiwan; chenyich@nchu.edu.tw
[2] Department of Industrial Engineering and Management, National Yunlin University of Science and Technology, Douliou, Yunlin 64000, Taiwan; b10921035@yuntech.edu.tw (C.-Y.W.); b10921137@yuntech.edu.tw (S.-P.T.); b10921141@yuntech.edu.tw (Y.-L.H.); b10921151@yuntech.edu.tw (R.-P.J.); b10721028@yuntech.edu.tw (T.-Y.Y.); wtchen@yuntech.edu.tw (W.-T.C.)
* Correspondence: d112056004@mail.nchu.edu.tw; Tel.: +886-0912577808
† Presented at the 31st International Conference on Geoinformatics, Toronto, ON, Canada, 14–16 August 2024.

Abstract: Natural disasters often occur unexpectedly, catching people off guard, such as the Hualien earthquake on 3 April 2024. Many of the evacuation-monitoring systems currently in place lack real-time updates of shelter capacities, which raises the risk of overcrowding under wartime scenarios. This study developed a system for the targeted assignment of evacuation sites during air raids. The DBSCAN algorithm was used to group data based on pedestrian flow patterns and an LSTM model was used to enhance the prediction speed and accuracy. Weighted Voronoi diagrams delineated regions to identify optimal evacuation points, while real-time SMS notifications through base station positioning disseminated evacuation information to the public. The experiment results demonstrated the effectiveness of the proposed system in facilitating safe evacuations across a broad range of geographic regions while reducing the number of LSTM models. Dynamic updates on the shelter capacities make it possible for citizens to make informed decisions during air raid emergencies.

Keywords: air raid evacuation; DBSCAN; LSTM; weighted Voronoi diagrams; disaster management

check for updates

Citation: Shih, C.-H.; Wu, C.-Y.; Tseng, S.-P.; Huang, Y.-L.; Jhuang, R.-P.; Chen, Y.-C.; Yang, T.-Y.; Chen, W.-T. Dynamic Evacuation Shelter Allocation in Response to Human Mobility: A Case Study of Taipei City. *Proceedings* **2024**, *110*, 32. https://doi.org/10.3390/proceedings2024110032

Dongmei Chen, Yuhong He and Songnian Li

Published: 18 February 2025

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The sudden onset and unpredictability of disasters, such as typhoons and earthquakes, means that the populace must be prepared and evacuation plans must be in place. Taiwan deals with potentially devastating typhoons from July to October, as well as earthquakes throughout the year. The island is subject to an average of 191 events of magnitude >4.0 on the Richter scale and 24 events of magnitude >5.0. The 921 earthquake (magnitude 7.6) prompted the adoption of annual earthquake evacuation drills. Urban populations must also deal with the risk of fire. Increased mobility and fluctuations in human traffic patterns can hinder the allocation of spaces at evacuation points. It is important to consider that most existing evacuation methods (e.g., disaster prevention cards or evacuation apps) are static in nature, which means that they cannot account for real-time crowd conditions, thereby increasing the risk of overcrowding. Figure 1 presents the operational procedures to be considered in the event of an earthquake.

In the current study, we developed a data-driven approach to the formulation of evacuation procedures. Our novel conceptual framework facilitates the tailoring of evacuation procedures according to local conditions and circumstances while improving the speed and accuracy of predictions. The proposed system uses a novel DBSCAN algorithm [1–7] for the clustering of crowd data based on similarity, as well as an LSTM [8–12] to predict crowd movements in each distinct cluster. A weighted Voronoi diagram is then used to

partition evacuation areas based on crowd volumes. The provision of multiple evacuation points within each area also enhances reliable access to shelters.

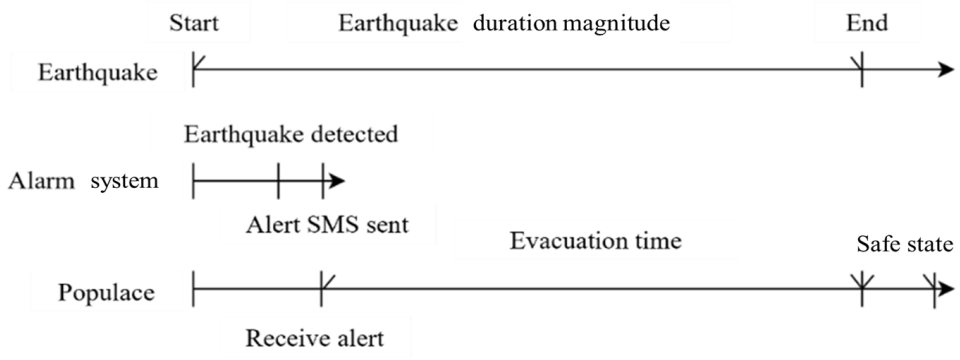


Figure 1. Conventional evacuation procedure for earthquakes.

2. Datasets

This study employed crowd flow data and shelter location data pertaining to Taipei City for the assignment of optimal shelter points. The crowd flow data included the crowd count, geographical coordinates of areas, and timestamps for the years 2018 and 2019. Shelter locations were selected based on data pertaining to air raid shelters, including shelter point coordinates and capacities collected from 1 to 31 March 2023.

3. Methods

Figure 2 presents the research flowchart for this study, the steps of which are addressed in the subsequent sections. The research methodology was divided into offline and online processing phases. The offline processing involved cleaning and transforming the crowd flow data before using the dynamic time-warping (DTW) algorithm to calculate the similarities in the crowd flow variations. DBSCAN clustering was then used to characterize the crowd patterns in each area, and a discrete LSTM model was trained to predict the crowd flow for each cluster. Online processing involved the use of weighted Voronoi diagrams to partition the region into separate evacuation areas based on the predicted crowd flow numbers with a different weight assigned to each point. The Voronoi diagrams were meant to facilitate the identification of suitable evacuation locations. Finally, we employed real-time location-based services (LBS) using base station positioning to send SMS alerts directing people to designated evacuation locations.

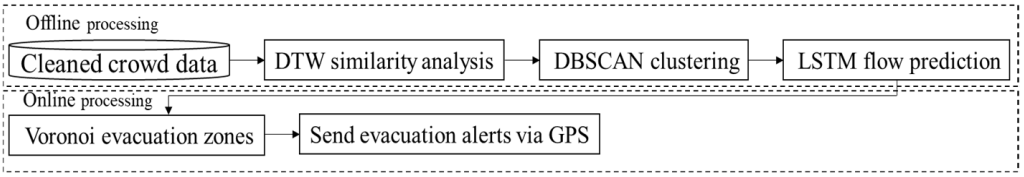


Figure 2. Research flow chart.

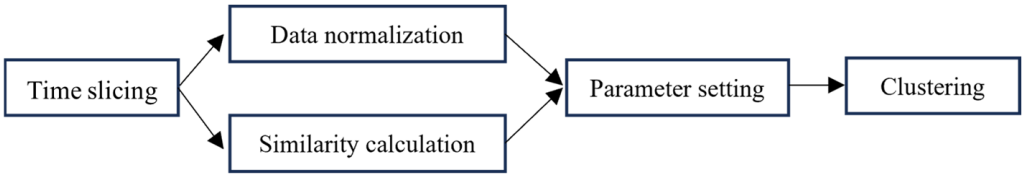


Figure 3. Clustering process diagram.

Figure 3 illustrates the process of similarity clustering by the DBSCAN algorithm. Step 1: Time slicing involved dividing each 24 h day into 8 intervals of 3 h and then calculating the average crowd count for each. Step 2: Data normalization of the crowd count for each time interval was then performed to increase the computation speed, accuracy, and stability of the DTW model used to calculate the crowd flow similarity. Normalization involved scaling the original data proportionally to a range of [0, 1] while preserving the original distribution, as follows:

$$x' = \frac{x - min(x)}{max(x) - min(x)}$$

where x is the original value, x' is the normalized value, $min(x)$ is the minimum value of the original dataset, and $max(x)$ is the maximum value of the original dataset.

Step 3: DTW was used to calculate the degree of similarity between pairs of data points. The lengths of the time series were the same so that the flow counts of sequences A and B were represented as $A = \{a_1, a_2, \ldots, a_n\}$ and $B = \{b_1, b_2, \ldots, b_n\}$.

Let D be an n*n matrix, where $D(I, j)$ represents the distance between the ith value of sequence A and the jth value of sequence B when aligned. Then, we used the formula to calculate and fill matrix D row by row or column by column, where $0 < i \leq n$ and $0 < j \leq n$. The final calculated distance was the value of $D(n, n)$, derived as follows:

$$|A(i) - B(j)| + (D[i-1,\ j-1],\ D[i-1,\ j],\ D[i,\ j-1].)$$

Note that smaller DTW values indicate a greater similarity between two points.

Step 4: We calculated the first quartile of all DTW matrix values and set the clustering EPS parameter to 210 and the minimum number of points to 2, indicating that at least two points were required to form a cluster. We then fine-tuned the parameters through experimental simulations to optimize the results.

Step 5: Clustering was performed in accordance with the optimized parameters. An example of clustering results is shown in Figure 4, in which the colored points indicate points of crowding based on time: (red) between the late afternoon and the early morning, (blue) between the afternoon and the late evening, (brown) between the morning and the afternoon, and (green) other.

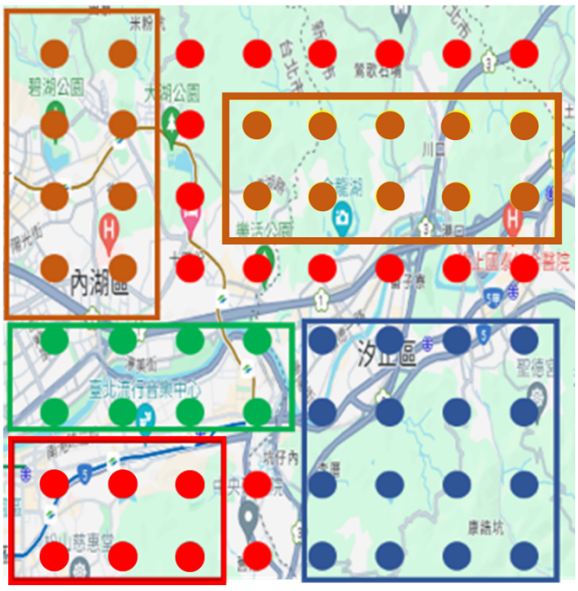


Figure 4. Schematic diagram of crowd flow clustering.

The LSTM was first trained for zone-based crowd prediction and then validated using the methods shown in Figure 5. Step 1: we imported the cleaned and clustered data for 2018. Step 2: we built an LSTM model and set the parameters, including the number of layers and the look-back value. Step 3: we trained the LSTM model using data for 2018. Step 4: we validated the model using 2019 data. Step 5: We observed the degree of discrepancy between the predicted and actual values (error rates). If the error was unreasonably high, we adjusted the batch size and retrained the model. This was reiterated until the error rate was reasonable. Step 6: we completed the LSTM prediction model. Note that for the sake of simplicity and efficiency in handling the time-series crowd flow data, we employed a single-layer LSTM without variables (e.g., weather or terrain). Crowd flow data from 2018 was used to make predictions for 2019, and actual crowd flow data from 2019 was used as the ground truth for validation using the mean absolute percentage error (MAPE), calculated as follows:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_1 - y_2}{y_1}\right|$$

where y_1 indicates the predicted value, y_2 indicates the actual value, and n indicates the population size. Note that the MAPE value is inversely proportional to the accuracy of the model, which means that any value exceeding this threshold will necessitate further refinement. In this study, we set a threshold MAPE of 10 as an acceptable error range.

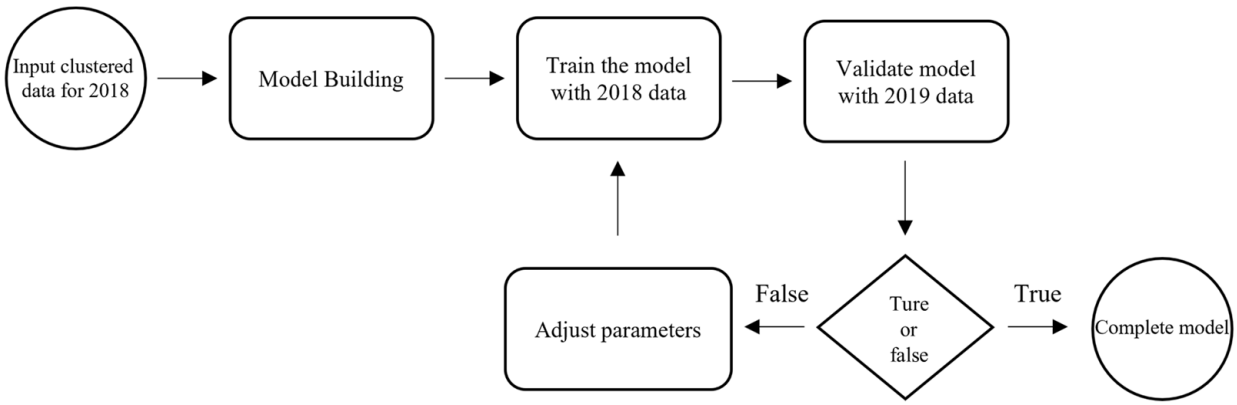


Figure 5. Training process of LSTM model.

Due to the nature of the crowd flow data used in this study, it was not possible to use standard Voronoi diagrams for the partitioning of the evacuation points. Instead, we opted for weighted Voronoi diagrams, wherein each point was assigned a weight based on the predicted number of people in a given area. Essentially, the weight determined the influence range of each point, i.e., the area affected by each Voronoi unit. Our ultimate goal was to provide shelter for as many people as possible by optimizing the assignment of evacuation points. This was achieved by varying the weight assigned to each point in accordance with the number of people predicted to enter that area.

In setting the weights, we assigned larger weights to areas with higher crowd flow, which resulted in some relatively large segmented areas with a correspondingly larger number of evacuation points. The assignment of lower weights to areas with lower crowd flow resulted in some relatively small areas with a limited range of movement for evacuation.

The weight parameters were fine-tuned using simulations. This involved adjusting the weights to prevent the crowd flow exceeding the total capacity of evacuation points in a given area, as follows:

$$Weight = \left(\frac{The\ number\ of\ people\ flowing\ at\ that\ point}{The\ total\ number\ of\ people\ flowing\ at\ all\ points}\right) \times \beta$$

where β was set to 100 to proportionally amplify the weight values for the sake of readability.

Weighted Voronoi partitioning illustrations were then integrated with real-time location data to direct people to designated evacuation points via SMS. Our objective was to ensure that everyone would be able to enter an evacuation point. Thus, we assessed the proposed system in terms of its ability to provide a space for everyone in a given area. This was achieved by running simulations using a set of variables established for area j during time period i, including $a_{i,j}$ (the capacity of an evacuation point), $b_{i,j}$ (the number of evacuees), and $k_{i,j}$ (the supply of evacuation points). We considered the entire year, which comprised 8760 time periods, each of which contained 433 points, which resulted in a total of 3,793,080 areas.

An $a_{i,j}$ value exceeding $b_{i,j}$ indicates that there is sufficient shelter space in that area, and thus, $k_{i,j}$ is set to 0. An $a_{i,j}$ less than $b_{i,j}$ indicates that there is insufficient shelter space in that area, and thus, $k_{i,j}$ is set to 1. This process was sequentially applied to every area, the accumulated results of which were used for validation.

Model building:

$$Min\ W = \frac{\sum_{i=1}^{8760}\sum_{j=1}^{433} k_{i,j}}{3,793,080} * 100\%$$

$$s.t. \qquad a_{i,j} \geq 0\ ,\ \forall i,j$$

$$b_{i,j} \geq 0,\ \forall i,j$$

$$k_{i,j} \epsilon \{0,1\},\ \forall i,j$$

$$i,j \geq 0$$

The formula used to evaluate the models in terms of space assignment was as follows:

$$\text{Total performance evaluation} = 1 - W$$

Our objective was to derive a minimal W solution, where W refers to the proportion of areas with insufficient shelter space. Note that a large W value indicates poorer overall performance.

4. Results and Discussion

Following multiple simulations and parameter adjustments, the optimal parameter settings were as follows: EPS (160) and min_Pts (2). As shown in Figure 6, the 433 crowd flow points for the entire year were divided into 12 clusters with 120 points of noise.

The clustering results were then used to establish 12 discrete LSTM crowd flow prediction models for each cluster, as well as one LSTM crowd flow prediction model for each noise point, for a total of 132 models. Note that each model used a single hidden layer. The simulations were used to compare the shelter capacities before and after the use of the weights for segmentation of the evacuation area. This resulted in the following three scenarios: (1) Weighted segmentation reduced the number of areas with insufficient shelter capacity. (2) Weighted segmentation had no effect on the number of areas with insufficient shelter capacity. (3) Weighted segmentation increased the number of areas with insufficient shelter capacity. Figure 7 presents an example Voronoi Diagram showing the Da'an District at 3:00 a.m. on 11 October 2019 (121.55, 25.025). This example presented four areas with insufficient evacuation point capacity. Figures 8 and 9, respectively, present Voronoi diagrams showing the segmentation of evacuation areas with and without a graph overlay. Here, the number of areas with inadequate shelter capacity decreased to three, thereby indicating the benefits of weighted segmentation. However, in the area with coordinates (121.555, 25.02), there were no evacuation points, regardless of whether weighted segmentation was or was not used. Following the segmentation of the evacuation areas using Voronoi diagrams and weighted Voronoi diagrams, we compared the capacity of evacuation points within each area versus the number of evacuees. We then separately input the data before and after using weights in the performance evaluation method.

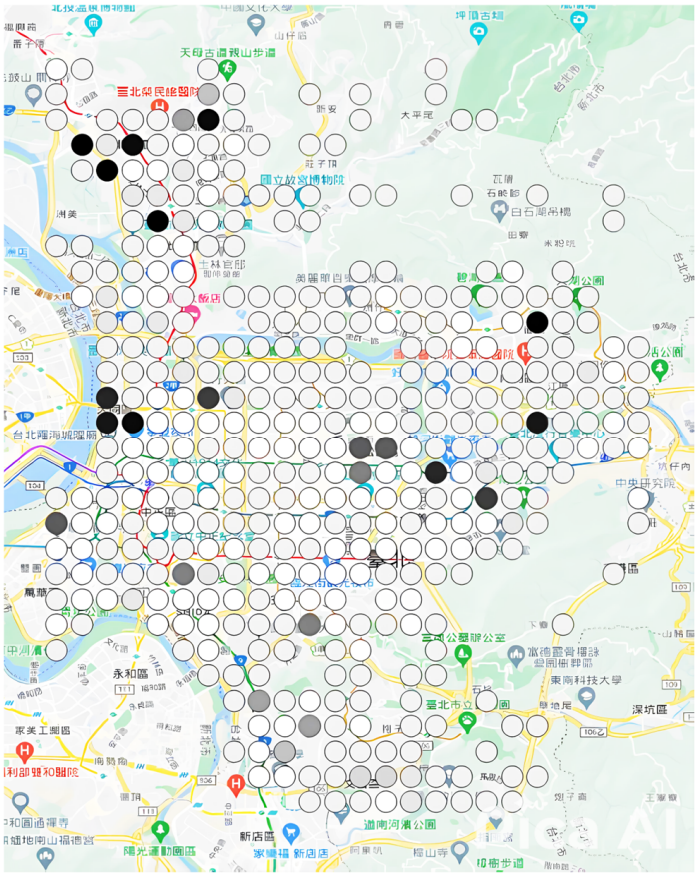


Figure 6. DBSCAN clustering diagram of crowd flow similarity over a year.



Figure 7. Voronoi diagram of a 3 × 3 grid in the Da'an District of Taipei at 3:00 a.m. on 11 October 2019.

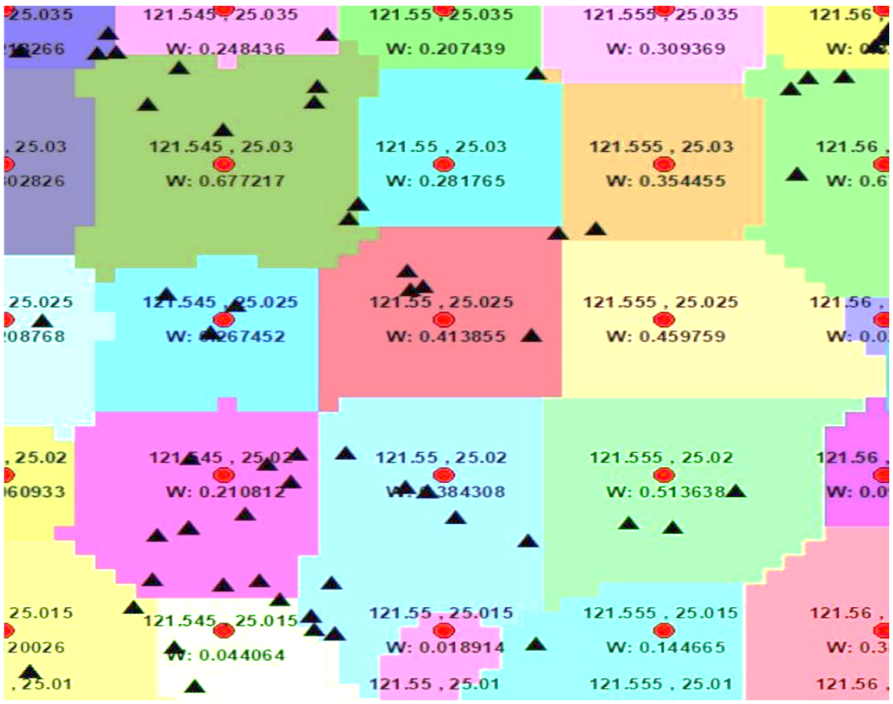


Figure 8. Voronoi diagram of the Da'an District, 11 October 2019, 3:00 a.m.

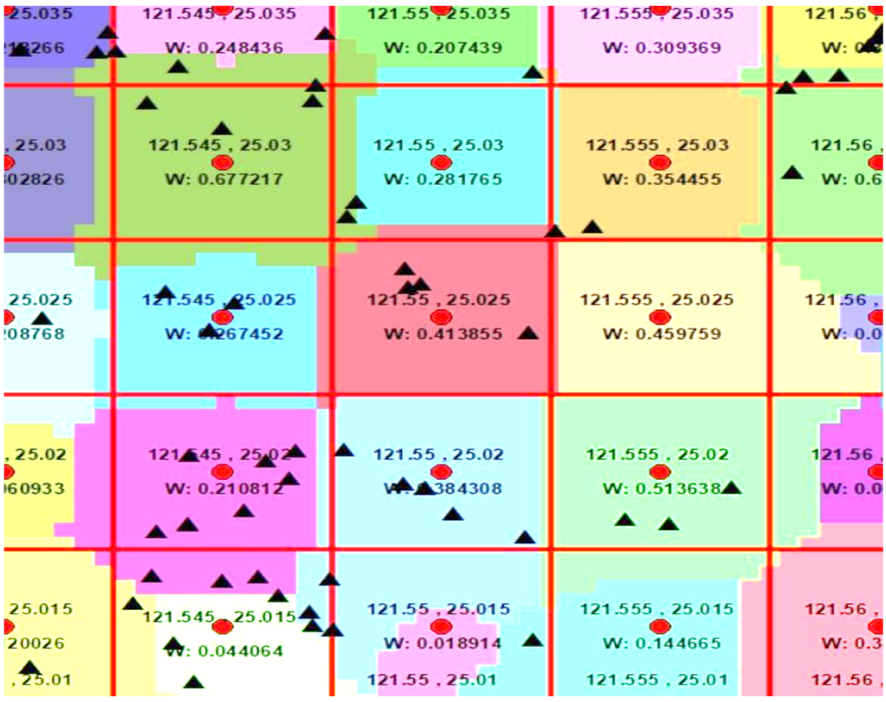


Figure 9. Evacuation segmentation overlay graph of the Da'an District, 11 October 2019, 3:00 a.m.

Segmentation without weights resulted in 2,382,720 areas with insufficient shelter capacity (i.e., performance = 37.182%). Segmentation with crowd flow weights resulted in 2,058,600 with insufficient shelter capacity (i.e., performance = 45.727%), which represented an improvement of 8.545%. This study devised a system aimed at ensuring that everyone has reliable access to evacuation shelters. This was achieved by clustering crowd flow data using DBSCAN and then establishing 132 LSTM models. The weighted Voronoi diagrams used in segmentation outperformed the conventional Voronoi diagrams in the allocation

of shelter space. The experimental results demonstrate the effectiveness of the proposed system in facilitating safe evacuations across a broad range of geographic regions while preserving the prediction accuracy using a limited number of LSTM models. Nonetheless, it was found that even the optimized assignment of evacuation shelters resulted in many areas with insufficient shelter capacity and a few areas with no shelters at all. We therefore recommend that in these areas, private buildings should be opened up to provide additional shelter space during emergencies.

5. Conclusions

The DBSCAN clustering method grouped similar crowd flow points, which led to the creation of 12 LSTM models for the clusters and 120 individual LSTM models for noise points, which totaled 132 models. The prediction results show that the errors were within an acceptable range, proving that DBSCAN effectively reduced the number of models, lowered the training costs, and maintained the prediction accuracy. The performance analysis showed that the use of weighted Voronoi diagrams resulted in a better overall performance compared with the traditional Voronoi diagrams. This indicates that the weighted division of evacuation spaces allows for a more efficient distribution of shelter capacity. However, the analysis also revealed that some areas lacked any shelters, while others had insufficient shelter capacity. While the weighted division improved the overall performance, the effect was limited. Therefore, it is recommended that during special periods, private buildings in these areas be opened for additional shelter space.

Author Contributions: Conceptualization, C.-H.S. and Y.-C.C.; methodology, C.-H.S.; validation, C.-Y.W., S.-P.T. and Y.-L.H.; formal analysis, R.-P.J.; investigation, T.-Y.Y.; resources, W.-T.C.; data curation, C.-Y.W.; writing—original draft preparation, C.-H.S.; writing—review and editing, C.-H.S.; visualization, S.-P.T.; supervision, Y.-C.C. and W.-T.C.; project administration, T.-Y.Y.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Yin, L.; Hu, H.; Li, K.; Zheng, G.; Qu, Y.; Chen, H. Improvement of DBSCAN Algorithm Based on K-Dist Graph for Adaptive Determining Parameters. *Electronics* **2023**, *12*, 3213. [CrossRef]
2. Hu, Y.; Li, J. A fast parallelized DBSCAN algorithm based on OpenMP for large-scale scientific data. *Front. Big Data* **2023**, *6*, 1292923.
3. Zhang, X.; Wang, Y.; Li, Z. Improved Earthquake Clustering Using a Density-Adaptive DBSCAN Algorithm. *Seismol. Res. Lett.* **2023**, *95*, 942–950.
4. Deng, C.; Li, Y. DBSCAN Clustering Algorithm Based on Density. *J. Comput. Sci. Technol.* **2023**, *38*, 812–823.
5. Kumar, S.; Sharma, A.; Singh, R.P. Enhanced DBSCAN Algorithm for Spatial Clustering in Big Data. *IEEE Access* **2023**, *11*, 87654–87667.
6. Khan, M.A.; Dutta, J.P.; Ryu, K.H. A Dynamic Density-Based Clustering Algorithm for Spatial Data Mining. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 3495–3509.
7. Wang, T.; Li, B.; Zhao, X. Optimized DBSCAN Algorithm for High-Dimensional Data Clustering. *IEEE Trans. Cybern.* **2024**, *54*, 1134–1147.
8. Zhu, W.; Tao, T.; Yan, H.; Yan, J.; Wang, J.; Li, S.; Xin, K. An optimized long short-term memory (LSTM)-based approach applied to early warning and forecasting of ponding in the urban drainage system. *Hydrol. Earth Syst. Sci.* **2023**, *27*, 2035–2050. [CrossRef]
9. Ghojogh, B.; Ghodsi, A. Recurrent Neural Networks and Long Short-Term Memory Networks: Tutorial and Survey. *arXiv* **2023**, arXiv:2304.11461.
10. Elsayed, N.; ElSayed, Z.; Maida, A.S. LiteLSTM Architecture Based on Weights Sharing for Recurrent Neural Networks. *arXiv* **2023**, arXiv:2301.04794.

11. Pasinato, H.D.; Reh, N.F.M. Modeling Turbulent Flows with LSTM Neural Network. *arXiv* **2023**, arXiv:2307.13784.
12. Ahmadi, A.M.S.; Semnani, S.H. Human trajectory prediction using LSTM with Attention mechanism. *arXiv* **2023**, arXiv:2309.00331.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.