IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 10 April 2024, accepted 24 April 2024, date of publication 29 April 2024, date of current version 20 May 2024.
Digital Object Identifier 10.1109/ACCESS.2024.3394905

RESEARCH ARTICLE

Crowd Congestion Forecasting Framework Using Ensemble Learning Model and Decision Making Algorithm: Umrah Use Case

ABDELOUAHID DERHAB[1], **IRFAN MOHIUDDIN**[2], **WALEED HALBOOB**[1], **AND JALAL ALMUHTADI**[1,2]
[1]Center of Excellence in Information Assurance (CoEIA), King Saud University, Riyadh 11451, Saudi Arabia
[2]College of Computer and Information Sciences, King Saud University, Riyadh 11461, Saudi Arabia

Corresponding author: Jalal Almuhtadi (jalal@ksu.edu.sa)

This work was supported by the Deputyship for Research and Innovation, Ministry of Education, Saudi Arabia, under Project IFKSUDR_D116.

ABSTRACT Forecasting crowd congestion is a critical aspect of crowd management, particularly in dynamic and densely populated areas, such as urban centers, events, or pilgrimage sites. In this paper, we proposed the first crowd congestion forecasting framework for the pilgrimage of Umrah. We addressed the crowd congestion forecasting problem by clustering the crowd flow trajectory in Masjid Al-Haram (Great Mosque) in the city of Makkah into six zones. The framework consists of two main components: 1) Ensemble forecasting model that aims at forecasting the crowd density of Masjid Al-Haram and its six zones, and 2) decision making algorithm that aims at keeping the crowd density at an acceptable level, and recommends *updating* the crowd flows when the forecasted crowd density exceeds the crowd density threshold. We built the ensemble learning model in three phases. In the first phase, we selected and evaluated different learning base models, including ARIMA, Sequence to Sequence (Seq2Seq) learning, M-1D-CNN-LSTM, and DeepSTN. In the second phase, the best three models, which performed well in the first phase, are selected to build the stacked ensemble model. The latter is validated using the walk-forward technique in the third phase. To evaluate the framework, we built a crowd dataset based on two temporal properties: 1) hourly context and 2) daily context. We evaluated the three phases of the ensemble forecasting model. In the first phase, DeepSTN performs the best by achieving a Mean Absolute Error (MAE) of 0.281. The results also indicate that DeepSTN is the best fit for five zones, and one variant of Seq2Seq, named Seq2Seq2b is the best fit for one zone under Mean Square Error (MSE) and Root Mean Squared Error (RMSE). Under MAE, DeepSTN and Seq2Seq2b, each of which is the best choice for three zones. In the second phase, the stacked ensemble achieves a MAE of 0.257. In the third phase, the stacked ensemble model is validated using the walk forward technique, which allows to reduce the MAE to 0.253. Although this framework focuses on Umrah, it can be customized for other use cases that involve crowd congestion forecasting.

INDEX TERMS Crowd congestion, crowd density, decision making, forecasting, learning model, Umrah.

I. INTRODUCTION

Umrah is one of the largest and the most crowded gatherings of people in the world, where pilgrims travel from across the world to the holiest site in Islam, i.e., the Masjid Al-Haram in the city of Makkah in Saudi Arabia to perform a set of rituals, which involves: (a) performing Tawaf, i.e., circumambulating the Kaaba (sacred cubic-shaped building structure located at the center of Masjid Al-Haram in a counterclockwise direction seven times), and (b) completing Sa'i, which is the act of walking seven times between the Hills of Safa and Marwah. The pilgrimage of Umrah often involves dealing

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

with significant crowdedness. The crowd congestion occurs particularly during peak seasons, such as Ramadan month, which is the ninth month of the Islamic lunar calendar, when a higher number of Muslims perform Umrah [12]. Therefore, crowd management strategies are required to ensure the safety and well-being of the pilgrims. Crowd management is a multidisciplinary field that involves the planning, organization, and controlling of large gatherings of people. Crowd management could integrate technologies such as surveillance systems, crowd monitoring tools, and data analytics to respond to potential issues such as overcrowding, bottlenecks, or emergency situations.

Crowd congestion is the overcrowding of people in a particular area, often resulting in restricted movement and discomfort. It can result in safety concerns, such as the risk of accidents and stampedes. To effectively manage this issue, some measures like crowd control, real-time monitoring, and infrastructure redesign could be implemented, in order to improve safety, and ensure a smoother flow of people in crowded environments [48]. To effectively deal with crowd congestion, it is important to forecast the crowd movement and crowd density, in order to anticipate any overcrowding situation beforehand, which can help in preventing accidents and emergencies [22].

The authorities of Saudi Arabia have deployed many technological solutions to manage the crowds in Masjid Al-Haram, including surveillance cameras, crowd density sensors, and other monitoring tools to provide real-time data on crowd movements. This information allows authorities to identify potential congestion areas, and make informed decisions to optimize crowd flow. However, these solutions are designed to detect crowd congestion when it occurs. So, there is a need for a forecasting solution to predict congestion before its occurrence. In order to deal with crowd congestion, the Saudi authorities have developed mobile applications like the Nusuk app [4], which allows pilgrims to schedule their visits to Masjid Al-Haram at a specific booked time. As it is not possible sometimes to arrive at the exact booked time, the authorities allows the pilgrims who arrive within a time frame of some hours before and after the booked time to enter Masjid Al-Haram. However, a crowd congestion could happen in a scenario where a huge number of pilgrims coincidentally come during the same time frame. Therefore, it is important to monitor the crowd density inside Masjid Al-Haram and forecast future crowd density in order to decide how many pilgrims can be allowed to enter Masjid Al-Haram in order to avoid congestion.

To deal with the above issues, we proposed a crowd congestion forecasting framework for the pilgrimage of Umrah. Although many forecasting models have been proposed in the literature [20], [39], but to the best of our knowledge, our work is the first that focuses on the forecasting of crowd congestion in the context of Umrah. More specifically, the main contributions of our paper are the following:

- We provided a crowd congestion forecasting framework, which is based on clustering the crowd flow trajectory in Masjid Al-Haram into six zones, as in each zone the pilgrims follow specific movement and perform certain activities.
- We proposed an ensemble learning model for the crowd congestion forecasting consisting of three phases. In the first phase, we selected learning base models and we evaluated them separately for the total area of Masjid Al-Haram, and for each of the six zones. In the second phase, we select the best three models that performed well in the first phase to build a stacked ensemble learning model (or meta model). In the third phase, the meta model is validated using the Walk Forward technique [19].
- We proposed a decision making algorithm that is triggered when the forecasted crowd density exceeds the crowd density threshold. The algorithm recommends two types of decisions: (a) decreasing the crowd flow coming to the zone, and (b) increasing the crowd flow leaving the zone.
- We built a crowd dataset by considering two temporal properties: (a) hourly context and (b) daily context. We used this dataset to evaluate the three phases of the ensemble forecasting model by selecting different learning base models, including ARIMA, Sequence to Sequence (Seq2Seq) learning, M-1D-CNN-LSTM, and DeepSTN.

The rest of the paper is organized as follows: Section II presents related work. In Section III, we formally describe the crowd congestion forecasting problem. The methodology of the proposed crowd congestion framework is described in Section IV. In Section V, we describe the implementation of the approach and dataset generation. Section VI presents the evaluation results. Finally, Section VII concludes the paper.

II. RELATED WORK

In this section, we give an overview of the most relevant literature on (a) crowd management in Hajj (annual Islamic pilgrimage) and Umrah, and (b) crowd forecasting.

A. CROWD MANAGEMENT IN HAJJ AND UMRAH

Different crowd management solutions for Hajj and Umrah have been proposed [24]. They leverage different technologies, including computer vision, spatial computing, artificial intelligence, and crowd modelling and simulation.

Crowd modelling and simulation aims at simulating the movement of crowds to understand their behaviors. For instance, Abdelghany et al. [9] presented a microsimulation model designed for the multidirectional flow of crowds. The model employs a cellular automata discrete system, which is used to depict walkways and movement areas within Mataf (area immediately surrounding the Kaaba). Zainuddin et al. [53] studied the issue of entrance to

Mataf zone, especially when it is congested. They used SimWalk simulator, which is based on the Social Force Model, to simulate the movement of pilgrims within Mataf. Sarmady et al. [47] simulated the circular movement of pilgrims during Tawaf using a cellular automata model in order to analyze the throughput of Mataf zone.

Computer vision has been an active research in Hajj and Umrah, which aims at counting the number of pilgrims in different spatio-temporal zones, and processing the videos to detect and avoid potential congestion scenarios [14], [46], [52]. Hussain et al. [28], proposed a crowd density estimation for Masjid Al-Haram by leveraging computer vision and backpropagation neural network. Khan [33] employed computer vision for the identification of congestion in Masjid Al-Haram by partitioning the crowd video clips into several overlapping temporal segments of equal duration, and trajectories are derived from these segments. The congestion points are identified by computing oscillation maps from these trajectories. Khan et al. [34] used IP cameras, which are deployed in Masjid Al-Haram to estimate crowd density, detect crowd congestion, and dominant crowd flows. Khozium [35] proposed a decision support system to analyze the crowd flow and density. The data are obtained through thermal cameras, and are processed to determine crowd density. The decision support system assesses decisions such as road closures, road priorities, and the organization of pilgrims.

Spatial computing focuses on scheduling and rescheduling the crowd movement during Hajj [21], [32], [36], [49], as well as on evacuation strategies that are capable of managing emergency situations [8], [23], [40], [43], [45], [50], [51].

Artificial intelligence techniques have been leveraged to manage the crowds in Hajj and Umrah [16]. Almutairi et al. [13] proposed a crowd management framework to deal with crowd issues in the context of Hajj. The framework uses advanced technologies like Wireless Sensor Networks, Cloud and Fog Computing, IoT, Machine Learning, Digital Cities, and RFID, aiming to ensure real-time monitoring, efficient data processing, and intelligent analysis. Halboob et al. [26] proposed a crowd intelligence framework that uses anomaly rules to detect crowd accidents. Abalkhail et al. [7] investigated the application of artificial intelligence in crowd management during the Hajj, specifically examining the experience of the Kingdom of Saudi Arabia (KSA) for handling large crowds in other contexts and locations. Alaff et al. [10] presented a framework that extracts the dynamic features based on the optical flows, and uses Generative Adversarial Network (GAN) and transfer learning strategy to detect abnormal behaviors in massive crowds. Bhuiyan et al. [15] utilized a fully convolutional neural network (FCNN) for crowd analysis, specifically focusing on the classification of crowd density. By employing FCNN, the method can effectively monitor and analyze the crowds, and hence providing insights into the density levels of different areas. In [29], machine learning techniques are used to identify crowd congestion incidents during Hajj through collecting data and feeding them to decision tree algorithms and K-Nearest Neighbors to predict the possibility of stampedes. Albattah et al. [11] applied CNN on mapped picture data to classify the crowds into five levels of crowdedness ranging from heavily-crowded to normal. Based on these levels, alarms are sent to avoid reaching the crowd limit.

B. CROWD FORECASTING

Zhang et al. [56] constructed a hybrid pedestrian flow detection model by analyzing real data from major mobile phone operators in China, utilizing information from smartphones and base stations. By employing the Log Distance Path Loss (LDPL) model and Gaussian Progress (GP) techniques, they estimated pedestrian density and retrieved information from raw network data through supervised learning.

Li et al. [41] presented a deep spatio-temporal learning-forecasting approach for predicting pedestrian walking paths in crowds. The approach extracts displacement information from walking history and embeds it into a Long-Short Term Memory-based architecture.

Zhang et al. [55] proposed an approach, called ResNet, to forecast the inflow and outflow of the crowds in different regions of a city. ResNet uses a residual neural network framework to capture the temporal closeness, period, and trend characteristics of crowd traffic. Each property is modeled by a branch of residual convolutional units that capture the spatial properties of crowd flow. The model dynamically aggregates the outputs of these branches, assigning weights based on data, and combines them with external factors like weather and day of the week to predict the crowd traffic in each region.

Jin et al. [30] proposed a deep-learning approach, named STRCNs, to deal with the same issue, forecasting the inflow and outflow of the crowds in the regions of a city. STRCNs uses Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) to capture spatio-temporal dependencies in the crowd flows. The model consists of four components: Closeness, Daily Influence, Weekly Influence, and External Influence. Each component is modeled by a branch of recurrent convolutional networks, and the outputs of the branches are merged together. STRCNs was tested on two datasets: MobileBJ and TaxiBJ.

He et al. [27] developed two deep crowd flow prediction architectures, named P-GRU and P-DBT, which use a Gated Recurrent Unit (GRU) model and Deep Bi-LSTM model respectively. The architectures integrate the precipitation records, to forecast the crowd flow. The architectures are evaluated under taxi trajectory data and bike trajectory data in Chongqing city in southwest-central China.

Zhao et al. [58] applied Convolution Neural Network (CNN) based method to forecast the crowd flows. They extracted crowd flows from mobile flow records (MFRs). The method is shown to reduce the forecasting error by 28% to

77% when compared with traditional time series regression models.

Kothari et al. [38] focused on human trajectory forecasting in crowds by dividing human motions into discrete intents and scene-specific refinements. They used interpretable knowledge-based functions and neural network predictions to select discretized intents based on human motion rules and complex interactions. The proposed approach was tested under TrajNet++ dataset [37].

Zhang et al. [57] proposed an urban crowd flow prediction model, named FPM-geo, which integrates geographic characteristics to improve the accuracy of spatial distribution predictions. By incorporating proximity, functional similarity, and road network connectivity using a residual multigraph convolution network, they effectively capture spatial dependency relationships between regions.

C. GAP ANALYSIS

By studying the literature, we identified the following gaps:

- To the best of our knowledge, there is no work that considers crowd forecasting, and specifically crowd congesting forecasting in the context of Umrah.
- In the literature, the crowd forecasting models are applied on the total studied area, and do not consider its zones, especially when the zones are characterized by specific crowd movement patterns, as in the context of Umrah.
- In the literature, there is no decision making algorithm that considers the output of the forecasting models to decide on the inflow and outflow rates of the crowds within the zones during Umrah.

III. CROWD CONGESTION FORECASTING PROBLEM

In this section, we describe how the crowd flow trajectory is clustered, and the formulation of the crowd congestion forecasting problem.

A. CROWD FLOW TRAJECTORY CLUSTERING

The movement of pilgrims in Masjid Al-Haram follows a predefined trajectory, consisting of a set of zones. In each zone, the pilgrims follow specific movements and perform activities. Figure 1 shows the geofencing of Masjid Al-Haram, and we identify the following six zones the pilgrim goes through during Umrah.

- *Mataf Zone (Zone 1)*: It is the area where the pilgrims perform Tawaf, which is accessed by pilgrims from different doors. It involves circumambulating the Kaaba in Masjid Al-Haram seven times. Pilgrims walk counterclockwise around the Kaaba. In Figure 1, this zone is illustrated in green color.
- *Transit Zone (Zone 2)*: This is the area where the pilgrims go after finishing their Tawaf. They proceed to the area behind "Muqam Ibrhaim" ("Station of Ibrahim" in English. According to Islamic tradition, the site marks the spot where the Prophet Ibrahim stood while he and his son Isma'il (Ishmael) built the Kaaba), as shown in the figure, to pray, and drink "Zam Zam" water (sacred well) before heading to the next zone, i.e., the Safa Hill to perform the ritual of Sa'i, which involves walking seven times back and forth between the hills of Safa and Marwah.
- *Safa Hill Zone (Zone 3)*: Pilgrims climb up to the Safa Hill, stop to recite supplications and verses from the Quran.
- *Safa to Marwah Zone (Zone 4)*: The pilgrims descend from Safa Hill and walk towards Marwah Hill. It is recommended for men to jog during Sa'i between two markers known as "the two green posts".
- *Marwah Hill Zone (Zone 5)*: Upon reaching Marwah Hill, the pilgrims stop again to recite supplications.
- *Marwah to Safa Zone (Zone 6)*: The pilgrims descend from Marah Hill and return to Safa Hill.

B. PROBLEM FORMULATION

Each zone at the t^{th} time interval, denoted by Z_i^t, is characterized by the following four features:

- Number of pilgrims: It is denoted by N_i^t, and represents the number of pilgrims in Zone i at time interval t.
- Crowd density: It is denoted by δ_i^t, and represents the crowd density of zone i at time interval t, which is obtained by dividing N_i^t by the size of the zone.
- Crowd inflow : It is denoted by In_i^t, and represents the number of crowds entering zone i in the t^{th} time interval.
- Crowd outflow: It is denoted by Out_i^t, and represents the number of crowds leaving zone i in the t^{th} time interval.

Formally, $Z_i^t = (N_i^t, \delta_i^t, In_i^t, Out_i^t)$. For each time interval t, we define the following:

- $Z^t = (Z_1^t|Z_2^t|Z_3^t|Z_4^t|Z_5^t|Z_6^t)$ is the crowd map that is captured at time interval t, where "$|''$ is the concatenation operator.
- $\delta^t = (\delta_1^t|\delta_2^t|\delta_3^t|\delta_4^t|\delta_5^t|\delta_6^t)$ is the crowd density observation that is captured at time interval t, where "$|''$ is the concatenation operator.

We also define the following:

- $Z = \{Z^1|Z^2|\cdots|Z^{n-1}\}$ as the historical crowd maps of Z_i^t from $t = 1$ to $t = n-1$.
- $\delta = \{\delta^1|\delta^2|\cdots|\delta^{n-1}\}$ as the historical crowd density observations of δ_i^t from $t = 1$ to $t = n-1$.

For any historical observation $Z = \{Z^1|Z^2|\cdots|Z^{n-1}\}$, the objective it predict the following:

- $\overline{Z^n}$, which is the predicted vector of $Z^n = (Z_1^n|Z_2^n|Z_3^n|Z_4^n|Z_5^n|Z_6^n)$, i.e., crowd map at time n
- $\overline{\delta^n}$, which is the predicted vector of $\delta^n = (\delta_1^n|\delta_2^n|\delta_3^n|\delta_4^n|\delta_5^n|\delta_6^n)$, i.e., crowd density vector at time n.

IV. METHODOLOGY

In this section, we present an overview of the crowd congestion forecasting framework, and its two main components: Ensemble learning model, and Decision making algorithm.

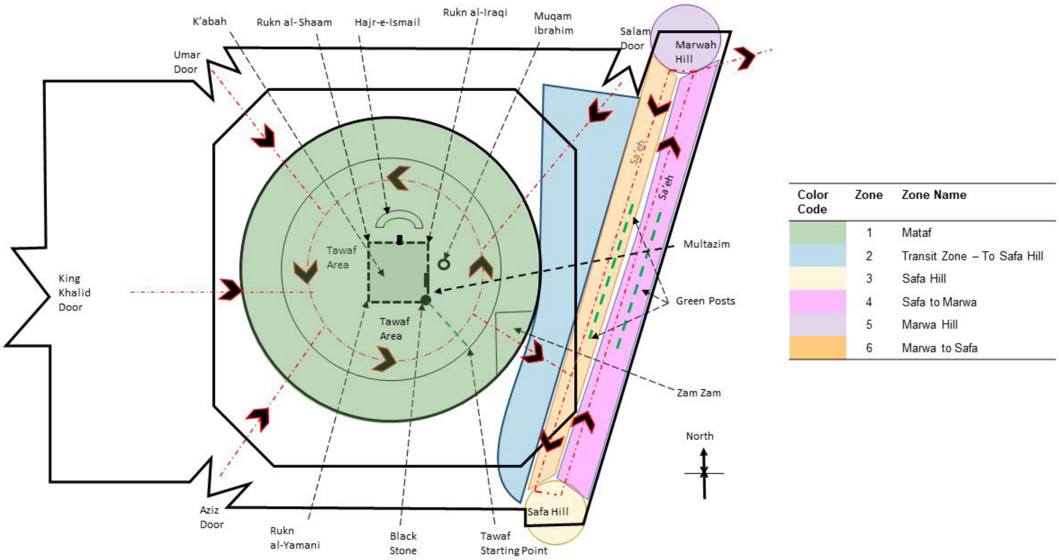

FIGURE 1. Geofencing of masjid al-haram in makkah.

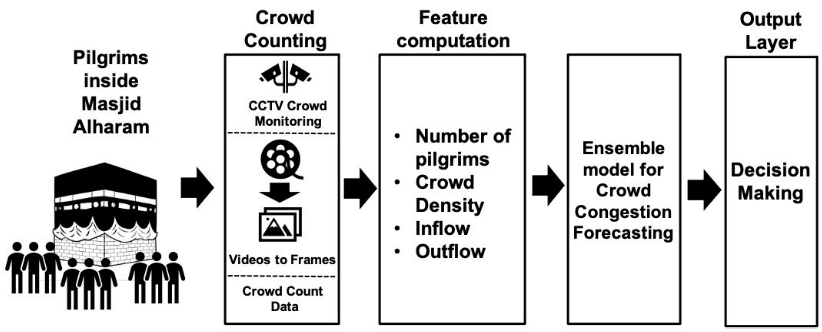

FIGURE 2. The proposed approach overview.

A. APPROACH OVERVIEW

Figure 2 shows the architecture of the forecasting framework. It consists of four main components:

- *Crowd counting*: The CCTV cameras that are deployed in Masjid Al-Haram capture frames. Object detection algorithms are used to detect human shapes and track their movements. After that, a counting algorithm is applied to keep track of the number of people detected in the frames. This algorithm should increment the count when a person enters the monitored zone and decrement it when someone exits.
- *Feature computation*: Based on the number of people per zone per time interval, which is obtained by the *Crowd counting*, the rest of features, i.e., crowd density, crowd inflow, and crowd outflow, are computed.
- *Ensemble learning model for crowd congestion forecasting*: It aims to forecast the crowd density of Masjid Al-Haram area and its zones for the next time interval. To do so, we follow an approach of three phases:
  - *Phase 1 (Base models)*: We apply different crowd forecasting models on Masjid Al-Haram area and its zones. In this phase, we identify the model that best fits each zone, i.e. the model that gives the lowest forecasting error of crowd map or crowd density. We also identify the models that incur the lowest forecasted crowd density for Masjid Al-Haram area.
  - *Phase 2 (Stacked ensemble model for crowd congestion forecasting)*: The best three models that are identified in *Phase 1* are stacked, as explained in Section IV-C, to generate the meta model, as shown in Figure 3.
  - *Phase 3 (Walk Forward Validation)*: We apply the Walk Forward validation technique on the meta model of *Phase 2*, as explained in Section IV-D to iteratively update the meta model, and make the final forecast, as shown in Figure 4.
- *Decision making algorithm*: It notifies the decision makers when a forecasted crowd density of a given zone

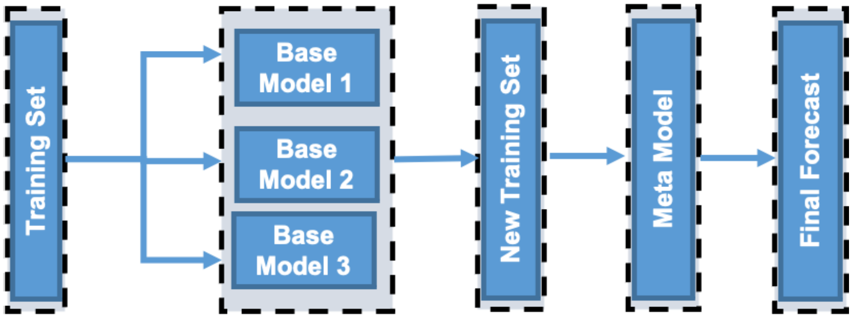

FIGURE 3. Stacked ensemble model for crowd congestion forecasting.

exceeds the crowd density threshold. It also suggests the updated crowd inflow and outflow rate to apply in order to keep the crowd density at an acceptable level.

B. BASE MODELS FOR CROWD CONGESTION FORECASTING

In this section, we describe the different base models for crowd congestion forecasting, which include ARIMA, Sequence to Sequence (Seq2Seq) learning, M-1D-CNN LSTM (Multiple one dimensional CNN-LSTM), and Deep-STN algorithm [25], [42], as shown in Figure 3. The selected base models are known for their usage in different time series forecasting problems [17], [18], [31], [54]. In addition, DeepSTN [25], [42] has been proven to perform well in the crowd density forecasting problem. The approach trains all the earlier-mentioned learning algorithms on all the six zones, and generates their corresponding testing models. For each zone, we select the model that incurs the lowest forecasting error.

1) ARIMA

We implement AutoRegressive Integrated Moving Average (ARIMA) model for each zone Z_i separately. ARIMA model is a time series forecasting method used to predict future values based on past observations and consists of two main components: AutoRegressive (AR), and Moving Average (MA). ARIMA model is denoted by $ARIMA(p, d, q)$, where p is the order of the AutoRegressive (AR) model, d is the order of differencing, i.e., the number of differences needed to make the time series stationary and required by the time series to get stationary, and q is the order of the Moving-Average (MA) model.

AutoRegressive (AR) The AR component represents the autoregressive part of the model, which is based on the idea that the future value of a time series can be predicted from its past values. For each zone Z_i, AR is expressed as follows:

$$AR(p) : \delta_i^t = c + \phi_1 \delta_i^{t-1} + \phi_2 \delta_i^{t-2} + \cdots + \phi_p \delta_i^{t-p} + \varepsilon_t$$

where δ_i^t is the crowd density of zone i at time t, c is an intercept term, and ε_t the white noise at time t.

ϕ_1, ϕ_2,$\cdots$,ϕ_p are the autoregressive coefficients for lag terms from 1 to p.

Moving Average (MA)

The MA component models the relationship between the current value and the past forecast errors (residuals). It is represented as follows:

$$MA(q) : \delta_i^t = c + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \cdots - \cdots - \theta_q \varepsilon_{t-q}$$

where δ_i^t is the crowd density of zone i at time t, c is an intercept term, and ε_t the white noise at time t.

θ_1, θ_2, $\cdots$ θ_q are the moving average coefficients for lag terms from 1 to q.

2) M-1D-CNN-LSTM

1D Convolutional Neural Network (1D-CNN) connected to a Long Short-Term Memory (LSTM) network is a deep learning architecture commonly used for sequence data analysis. We consider a neural network architecture, as shown in Figure 5, that combines multiple 1D-CNNs, connected to LSTM, as follows:

1) Multiple 1D-CNNs: We employ six 1D CNNs that are executed in parallel and each one takes as input the vector feature of a specific zone, Z_i. Each 1D-CNN outputs a feature map specific to the patterns it is designed to forecast.
2) Concatenation of CNN outputs: The output feature maps from the individual CNNs are concatenated into a single feature vector.
3) LSTM: The concatenated feature vector is then passed to an LSTM network.
4) Fully Connected Layers and Output: Following the LSTM network, we can find fully connected layers for higher-level feature representation and prediction. The final layer typically consists of ReLu activation function to produce the output of the model, which is the forecasted vector $\overline{\delta^n}$.

3) SEQUENCE TO SEQUENCE LEARNING MODEL

Sequence-to-sequence (Seq2Seq) learning is a deep learning architecture used for tasks that involve transforming an input sequence into an output sequence. The Seq2Seq architecture, as shown in Figure 6, consists of the following components:

- Encoder: It is an LSTM encoder that takes the input sequence and processes it sequentially, typically one

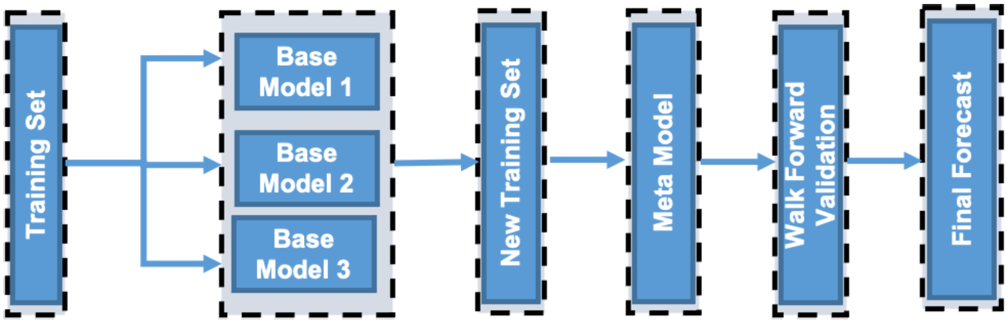

FIGURE 4. Stacked ensemble model for crowd congestion forecasting with walk forward validation.

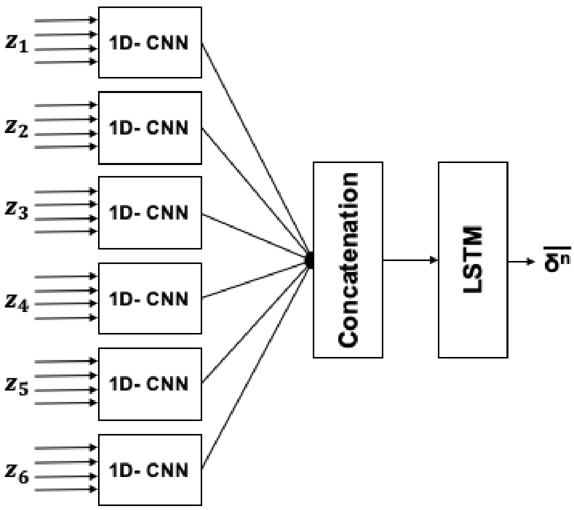

FIGURE 5. The utilized M-1D-CNN-LSTM architecture.

token at a time. The encoder's recurrent or convolutional layers transform these input vectors into a fixed-size context or a hidden state vector, which captures the information from the entire input sequence. This context vector serves as the summarization of the input information.

- Context vector: It is the intermediate representation of the input sequence that contains relevant information about the sequence. It captures the semantic and contextual information of the input sequence, which is necessary for generating the output sequence.
- Decoder: It is an LSTM decoder that takes the context vector produced by the encoder, and generates the output sequence token by token, with each token's probability distribution being conditioned on the previous tokens and the context vector.

Table 1 describes the different variants of Seq2Seq architecture that are used in this work.

4) DEEPSTN

DeepSTN is a deep learning architecture that is proposed in [25] and [42] to predict crowd flows in the metropolis. The crowd flow is modeled using three temporal features, named (a) Closeness, (b) Period, and (c) Trend, corresponding to the recent time intervals, daily periodicity, and weekly trend respectively. In our case, we apply DeepSTN on the crowd of Masjid Al-Haram with more focus on predicting crowd density instead of crowd flow as in DeepSTN. Figure 7 depicts DeepSTN architecture, which is structured as follows:

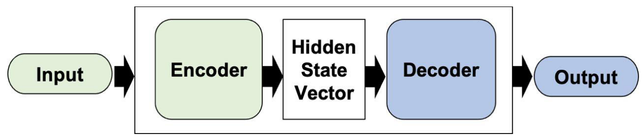

FIGURE 6. The utilized Seq2Seq architecture.

- 2D convolution: The input feature, i.e., vector Z is fed to three 2D convolutions, representing the Closeness, Period, and Trend properties.
- Concatenation: The outputs of the three 2D convolutions are concatenated to form a feature vector.
- 2D convolution: The concatenated vector is fed to 2D convolution to predict vector $\overline{Z^n}$.

C. STACKED ENSEMBLE LEARNING OR ENSEMBLE STACKING

Stacking, also referred to as stacked ensemble, is a powerful ensemble learning technique. The main advantage of the stacked ensemble learning is that it leverages the predictions of multiple base models to produce potentially more accurate final model that typically exhibits improved accuracy compared to individual base models. In addition, the stacked ensemble learning employs heterogeneous base models whereas bagging and boosting ensemble learning employ homogeneous base models [44]. The primary idea in our case study is to feed the predictions from base models to a higher level model known as Meta Model in order to obtain the final forecast of crowd congestion.

The process of stacked ensemble learning is as follows:

1) *Dataset splitting*: The dataset is partitioning into time series datasets: training, validation, and testing.

TABLE 1. Variants of Seq2Seq architecture.

Variant	Architecture	Input Sequence	Output Sequence
Seq2Seq1a	1 Encoder-1 Decoder	δ	$\overline{\delta^n}$
Seq2Seq1b	2 Encoder-2 Decoder	δ	$\overline{\delta^n}$
Seq2Seq2a	1 Encoder-1 Decoder	Z	$\overline{Z^n}$
Seq2Seq2b	2 Encoder-2 Decoder	Z	$\overline{Z^n}$
Seq2Seq3a	1 Encoder-1 Decoder	Z	$\overline{\delta^n}$
Seq2Seq3b	2 Encoder-2 Decoder	Z	$\overline{\delta^n}$

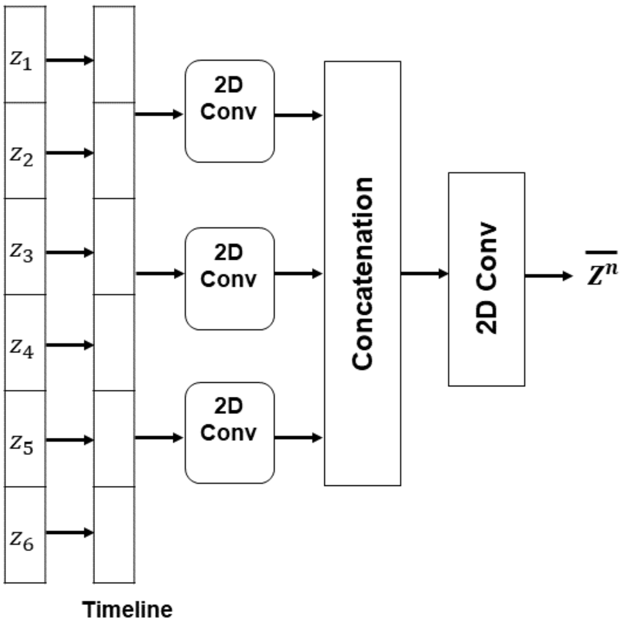

FIGURE 7. The utilized DeepSTN architecture.

2) *Base model selection*: We select a variety of forecasting base models suitable for time series prediction.
3) *Training the base models*: Each base model is trained on the training dataset.
4) *Forecasting on the validation set*: After training the base models, they are applied on the validation set separately to generate their forecasts of crowd congestion.
5) *Training the meta model*: a meta model, often referred to as a meta learner, is constructed. This meta model utilizes the forecasts generated by the base models on the validation set to train the meta model. These forecasts serve as input features (or new training dataset) for training the meta model. Various algorithms, such as linear regression, logistic regression, can be employed to construct this model. The meta model learns to combine the forecasts from the base models to make a final forecast for each future time step.
6) *Final forecasts*: The meta model is used to make forecasts for future time steps in the time series, which is the testing dataset. The forecasts made by the base models on the testing dataset are fed to the meta model, which then produces the final forecast for each time step.
7) *Model evaluation*: The performance of the stacked ensemble model is evaluated. This involves comparing the final forecasts made by the meta model with the actual values in the testing dataset, using appropriate forecasting metrics, such as mean squared error (MSE), or other metrics suitable for time series forecasting tasks.

D. WALK FORWARD VALIDATION

Walk Forward validation [19] is a time-series cross-validation technique, which is specifically designed to evaluate the performance of models used in time series forecasting. It involves iteratively training the model on training dataset, making forecasting for a subsequent test window of W time steps, and then updating the training dataset to include the W test data observations. The Walk Forward validation is integrated with the stacked ensemble learning model, and consists of the following steps:

1) *Initial selection*: We set W to a predefined value, and we define a *test window* to include the first W observations in the testing dataset.
2) *Model training*: We employ the training dataset to train the meta model, using the methodology explained in Section IV-C.
3) *Model testing*: We use the trained model to generate forecasts for the test window.
4) *Window sliding*: The training dataset is expanded to include the actual values of the test window, and the test window is shifted forward in time to the next W test observations.
5) *Iterative process*: We return to step (2) and repeat the process until we move through all the observations in the testing dataset.
6) *Performance evaluation*: The forecasts are evaluated against the actual values of all testing data, and the forecasting error metrics are aggregated from all testing data to evaluate the overall performance of the model.

E. DECISION MAKING

The decision making algorithm is triggered when the forecasted crowd density for the next time interval of any zone exceeds the crowd density threshold. The objective of the decision-making algorithm is to determine updated values for crowd inflow or crowd outflow in order to make the crowd density of the zone reaches the crowd density threshold, and does not exceed that threshold for the next time interval. To achieve this objective, we consider two types of decisions: (1) decreasing the crowd inflow entering the zone, or (2) increasing the crowd outflow leaving the zone. The question is: which decision to take for which zone? To answer this question, we need to examine the crowd patterns of the six zones. We can identify two types of crowd patterns:

- *Zones with continuous mobility pattern*: In Masjid Al-Haram, the pilgrims circumambulate Kaaba seven times. They also perform Sa'i, which involves walking seven times back and forth between the hills of Safa and Marwah. Thus, the zones: 1, 4, and 6 represent the areas where the pilgrims perform the Umrah rituals, and characterized by continuous mobility pattern. To have a smooth Umrah experience and avoid congestion in these zones, it important to reduce the flow rate of pilgrims entering these zones.
- *Zones with no or low mobility pattern*: In the transition zone (Zone 2), the pilgrims pray, drink "Zam Zam" water, and might stay to recite supplications and verses from the Quran before going to Safa Hill. So, this zone is characterized by low mobility models. In the Safa Hill and Marwah Hill zones (Zone 3 and Zone 5), the pilgrims stay standing to recite supplications. To avoid crowd congestion, it is important to make pilgrims who stay a long time in these regions to leave them and go to the next zone.

Based on the above-mentioned mobility patterns, Algorithm 1 describes their corresponding two decision making cases:

- Case A: It is triggered when the forecasted crowd density of Mataf (Zone 1), Safa to Marwah (Zone 4), or Safa to Marwah (Zone 6) exceeds the crowd density threshold (Algorithm 1: line 1), i.e, the flow of pilgrims that are entering these zones could cause congestion. Thus, we need to decrease the crowd inflow coming to these zones. The algorithm computes a new crowd inflow for the next time interval, which keeps the number of pilgrims inside the zones and its crowd density less than the capacity of the zones and the crowd density threshold, respectively (Algorithm 1: line 5). If the computed crowd inflow is less than the current one (Algorithm 1: line 6), an alert is sent to the crowd management personnel to manage the crowd flow, to decrease the flow of pilgrims coming from the doors of Masjid Al-Haram, Zone 3, or Zone 5 to the new crowd inflow value.
- Case B: It is triggered when the forecasted crowd density of Transition (Zone 2), Safa Hill (Zone 3) or Marwah Hill (Zone 5) exceeds the crowd density threshold (Algorithm 1: line 1o), i.e., the pilgrims are staying in these zones for a longer time and are not moving, which could cause crowd congestion. Thus, we need to increase the crowd ouflow leaving these zones. The algorithm computes a new crowd outflow for the next time interval, which keeps the number of pilgrims inside these zones and their crowd density less than their capacity and the crowd density threshold, respectively (Algorithm 1: line 14). If the computed crowd outflow is higher than the current one (Algorithm 1: line 15), an alert is sent to the crowd management personnel to manage the crowd flow, to increase the flow of pilgrims leaving Zone 2, Zone 3, or Zone 5 to the new crowd outflow value.

Algorithm 1 Decision Making Algorithm

INPUT:
N_In : Number of people entering a zone;
N_Out : Number of people leaving a zone;
Δ : Time interval of each observation;
δ_{th} : Crowd density threshold;
$N_{th}(i)$: Maximum amount of people that can be in Zone i;
New_Inflow : Updated crowd inflow;
New_Outflow : Updated crowd outflow;
CASE A: Forecasted crowd density of Mataf, Safa or Marwa exceeds the the crowd density threshold
1: **if** $\overline{\delta_i^n} > \delta_{th}$ **then** $\triangleright\ i \in \{1, 4, 6\}$
2: $\quad N\_In = In_i^{n-1} \times \Delta$;
3: $\quad N\_Out = Out_i^{n-1} \times \Delta$;
4: $\quad N_{th}(i) = \delta_{th} \times Size(Zone\, i) = N_i^{n-1} + N\_In - N\_Out$;
5: $\quad New\_Inflow = \frac{N_{th}(i) - N_i^{n-1} - N\_Out}{\Delta}$;
6: $\quad$ **if** $New\_Inflow < In_i^{n-1}$ **then**
7: $\qquad In_i^n = New\_Inflow$;
8: $\quad$ **end if**
9: **end if**
CASE B: Forecasted crowd density of Transition/ Safa Hill/ Marwah Hill exceeds the the crowd density threshold
10: **if** $\overline{\delta_i^n} > \delta_{th}$ **then** $\triangleright\ i \in \{2, 3, 5\}$
11: $\quad N\_In = In_i^{n-1} \times \Delta$;
12: $\quad N\_Out = Out_i^{n-1} \times \Delta$;
13: $\quad N_{th}(i) = \delta_{th} \times Size(Zone\, i) = N_i^{n-1} + N\_In - N\_Out$;
14: $\quad New\_Outflow = \frac{N_i^{n-1} + N\_In - N_{th}(i)}{\Delta}$;
15: $\quad$ **if** $New\_Outflow > Out_i^{n-1}$ **then**
16: $\qquad Out_i^n = New\_Outflow$;
17: $\quad$ **end if**
18: **end if**

V. IMPLEMENTATION

We use Python programming language, and TensorFlow to build the forecasting deep learning models of M-1D-CNN-LSTM and Seq2Seq. Moreover, different libraries are used including Scikit-learn, and numpy. To implement ARIMA, we use pmdarima statistical library [5] with Auto-Arima function to determine the optimal values of p, d, and q for ARIMA model that best fits the dataset, and we use the code in [1] to implement DeepSTN.

TABLE 2. Hourly context of the dataset.

Range of prayer times in 2023	Hour in a day	Hourly Crowd density level	Crowd range
Fajr=[4:10 AM, 5:40 AM]	[Isha, Fajr]	Very high	[150,200]
Dhuhr=[12:04 PM, 12:34 PM]	[Fajr, Dhuhr]	Medium	[30,60]
Asr=[3:16 PM, 3:54 PM]	[Dhuhr, Asr]	Low	[0,0]
Maghrib=[5:37 PM, 7:07 PM]	[Asr, Maghrib]	Medium	[30,60]
Isha=[7:07 PM, 8:37 PM]	[Maghrib, Isha]	High	[80,100]

A. DATASET GENERATION

We create a dataset of 1 year, i.e., the year 2023, where we generate a sample Z^t for each hour, which leads to a dataset consisting of 8760 samples. For each sample, the number of pilgrims per hour is generated based on two temporal contexts: hourly context and daily context.

In the hourly context, we define four levels of hourly crowd density: Very high, High, Medium, and Low. As the city of Makkah is known for its hot temperature, many pilgrims prefer to perform Umrah under low temperatures, and move to Masjid Al-harama especially during the prayer times. The crowd density is high between Maghrib prayer time and Isha prayer time, and becomes higher between Isha prayer time and Fajr prayer time. After that, the density becomes medium until Dhuhr prayer time. Between Dhuhr prayer time and Asr prayer time, the temperature is high, and the crowd density becomes low, and returns to be medium between Asr prayer time and Maghrib prayer time. The prayer times change throughout the year due to variations in daylight hours. In Table 2, we present the range of prayer times in 2023 [3].

For each crowd density level, we define a crowd range, which is a parameter of the *Random.randint* function [6], as defined in Table 2. We define a function, named $context\_per\_day(h)$ that returns the result of executing the *Random.randint* function.

The crowd density of pilgrims also depends on the characteristics of the day. We define the qualitative scale for the daily crowd density level, denoted by DC_i, $1 \leq i \leq 6$, and we define a function for each level, as shown in Table 3. These functions are used to define the *context* of the sample as in Equation 1, shown at the bottom of the page.

The daily crowd density level is defined as: $DC_i < DC_{i+1}$, $1 \leq i \leq 5$. It is known that the number of pilgrims during a week-end is higher than in a normal day. In Ramadan month (the ninth month of the Islamic lunar Hijri Calendar), this number largely increases, and particularly in the week-ends, in the last 10 days, and the odd days of the last 10 days of the month, and it reaches its maximum on the 26^{th} day of the month. To identify Ramadan days, we use hijri-converter python library [2] to convert Gregorian dates to Hijri ones.

After that, we compute the number of pilgrims in each zone, as presented in Table 4. The values of range parameter in *random.randint* function are chosen according to the size of the zones. For larger zones, we assign larger values to the range parameter and vice versa.

From Table 4, we derive the corresponding crowd density using the size of the zones in Table 5. The crowd inflow and crowd outflow of the zones are randomly set between 500 and 2000 persons per hour, as in Equation 2.

$$random.randint(500, 2000) \times (1 + context(h, d)) \tag{2}$$

The crowd density threshold is set to 4.5 persons/ square meter. The crowd count threshold per zone is obtained by multiplying the crowd density threshold by the size of the zone. The last 14 days from the generated dataset are chosen as testing data, and the rest as training data. Table 5 defines the following parameters for each zone:

- Size: It represents the size of the zone in square meters.
- Crowd count threshold: It represents the maximum allowable number of pilgrims in the zone.

VI. EVALUATION RESULTS

A. EVALUATION METRICS

To evaluate the performance of the forecasting models, we use the Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE) metrics:

$$MSE = \frac{1}{T}\sum_{i=1}^{T}(Y_i - \overline{Y_i})^T \tag{3}$$

$$RMSE = \sqrt{\frac{1}{T}\sum_{i=1}^{T}(Y_i - \overline{Y_i})^2} \tag{4}$$

$$MAE = \frac{1}{T}\sum_{i=1}^{T}|(Y_i - \overline{Y_i})| \tag{5}$$

where Y_i is the actual observation at the i^{th} time interval, $\overline{Y_i}$ is the predicted value of Y_i, and T is the size of the testing set.

The above performance metrics are evaluated under two cases:

$$context(h, d) = \frac{context\_per\_day(h) + is\_weekend(d) + is\_ramadan(d) + is\_weekend\_ramadan(d)+}{100} \\ \frac{is\_last\_10\_day\_ramadan(d) + is\_odd\_day\_ramadan(d) + is\_26\_ramadan(d)}{100} \tag{1}$$

TABLE 3. Daily context of the dataset.

Function	Description	Daily crowd density level
is_weekend(d)	Returns 1 if day d is a week-end, i.e, Friday, Saturday or national day	DC_1
is_ramadan(d)	Returns 1 if day d is in Ramadan month	DC_2
is_weekend_ramadan(d)	Returns 1 if day d is in Ramadan month and also a week-end	DC_3
is_last_10_day_ramadan(d)	Returns 1 if day d is in the last 10 days of Ramadan month	DC_4
is_odd_day_ramadan(d)	Returns 1 if day d is the 20^{th}, 22^{th}, 24^{th}, 26^{th}, or 28^{th} day of Ramadan month	DC_5
is_26_ramadan(d)	Returns 1 if day d is the 26^{th} day of Ramadan month	DC_6

TABLE 4. Number of pilgrims per zone.

Zone	Number of pilgrims	Computation function
Mataf	$N_1^{24\times d+h}$	$\mathrm{random.randint}(5000, 7500) \times (1 + \mathrm{context}(h, d))$
Transition	$N_2^{24\times d+h}$	$\mathrm{random.randint}(1500, 2000) \times (1 + \mathrm{context}(h, d))$
Safa Hill	$N_3^{24\times d+h}$	$\mathrm{random.randint}(500, 1000) \times (1 + \mathrm{context}(h, d))$
Safa to Marwah	$N_4^{24\times d+h}$	$\mathrm{random.randint}(4500, 6500) \times (1 + \mathrm{context}(h, d))$
Marwah Hill	$N_5^{24\times d+h}$	$\mathrm{random.randint}(500, 1000) \times (1 + \mathrm{context}(h, d))$
Marwah to Safa	$N_6^{24\times d+h}$	$\mathrm{random.randint}(4500, 6500) \times (1 + \mathrm{context}(h, d))$

TABLE 5. Parameters of the zones.

Zone	Size (m^2)	Crowd count threshold
Mataf	8000	36000
Transition	2000	9000
Safa Hill	1000	4500
Safa to Marwah	7000	31500
Marwah Hill	1000	4500
Marwah to Safa	7000	31500
Total	26000	117000

- *Global metrics*: In MSE, RMSE, and MAE metrics, Y_i and $\overline{Y_i}$ are set to δ^i and $\overline{\delta^i}$ respectively.
- *Per-zone metrics*: We compute the MSE, RMSE, and MAE metrics for each zone j by setting Y_i and $\overline{Y_i}$ to δ^i_j and $\overline{\delta^i_j}$ respectively.

B. CROWD FORECASTING RESULTS OF BASE MODELS

Table 6 shows the performance of the forecasting models under the *Global metrics*. The first observation we can draw from the table is that ARIMA incurs the highest forecasting errors compared to the deep learning algorithms. This can be explained by the fact that ARIMA is a statistical algorithm, which means it assumes a linear relationship between past and future values in a time series. Deep learning algorithms, on the other hand, are capable of capturing complex, non-linear patterns in the data. In addition, ARIMA relies on a fixed number of past observations, i.e., the order of autoregression and moving average terms, to make predictions. Deep learning algorithms, such as Long Short-Term Memory (LSTM), have the ability to capture longer-term dependencies in the data due to their recurrent nature.

We can also observe that DeepSTN incurs the lowest forecasting errors as it is designed to capture a more complex long-range spatial dependency between the crowd flows in the zones. It is based on the *temporal dependency*, which means that the crowd flows in a zone is affected by recent time intervals, and it defines three temporal properties: closeness, period, and trend. For each temporal feature, 2D convolution network is employed, and the outputs of the three convolutions are aggregated to be fed to another 2D convolution network.

Seq2Seq2b comes second after DeepSTN and both of them use the same input sequence and output sequence, which allows Seq2Seq2b to capture more dependencies in the data. Additionally, Seq2Seq2b employs two encoders and two decoders, which slightly offers better results than Seq2Seq2a that only employs one encoder and one decoder. The forecasting errors of Seq2Seq1a and Seq2Seq1b slightly increase compared to Seq2Seq2a and Seq2Seq2b, as their input sequence is δ, which is smaller than Z, and hence less dependencies in the data are captured. In Seq2Seq3a, Seq2Seq3b, and M-1D-CNN-LSTM, the size of the input sequence and output sequence are different, and they incurs the highest forecasting errors among the deep learning models.

In Table 7 and Table 8, we show the performance of the forecasting models under the *Per-zone metrics*. Under MSE and RMSE metrics, we can observe that DeepSTN is the best fit for all zones except for Marwah Hill. In case of Marwah Hill, Seq2Seq2b comes first with $MSE = 0.207$ and $RMSE = 0.455$ then Seq2Seq2a comes second with $MSE = 0.214$ and $RMSE = 0.462$, followed by DeepSTN with $MSE = 0.217$ and $RMSE = 0.466$. Under MAE, DeepSTN performs the best for three zones: Transit, Safa to Marwah, and Marwah to Safa. On the other hand, Seq2Seq2b provides the best results for Mataf, Safa Hill, and Marwah Hill zones.

C. CROWD FORECASTING RESULTS OF THE STACKED ENSEMBLE LEARNING MODEL

To evaluate the performance of the stacked ensemble learning model, we select the best three forecastng models that

TABLE 6. Forecasting results under global metrics.

Model	Input Sequence	Output Sequence	Y_i	$\overline{Y_i}$	MSE	RMSE	MAE
ARIMA	δ	$\overline{\delta}$	δ^i	$\overline{\delta^i}$	1.536	1.239	0.995
Seq2Seq1a	δ	$\overline{\delta}$	δ^i	$\overline{\delta^i}$	0.175	0.419	0.297
Seq2Seq1b	δ	$\overline{\delta}$	δ^i	$\overline{\delta^i}$	0.176	0.420	0.306
Seq2Seq2a	Z	$\overline{Z}$	δ^i	$\overline{\delta^i}$	0.17	0.413	0.297
Seq2Seq2b	Z	$\overline{Z}$	δ^i	$\overline{\delta^i}$	0.16	0.401	0.283
Seq2Seq3a	Z	$\overline{\delta}$	δ^i	$\overline{\delta^i}$	0.231	0.481	0.345
Seq2Seq3b	Z	$\overline{\delta}$	δ^i	$\overline{\delta^i}$	0.254	0.504	0.373
M-1D-CNN-LSTM	Z	$\overline{\delta}$	δ^i	$\overline{\delta^i}$	0.287	0.536	0.398
DeepSTN	Z	$\overline{Z}$	δ^i	$\overline{\delta^i}$	0.144	0.379	0.281

TABLE 7. Forecasting results under per-zone metrics (Part 1).

Model	Mataf			Transit			Safa Hill		
	MSE	RMSE	MAE	MSE	RMSE	MAE	MSE	RMSE	MAE
ARIMA	1.168	1.081	0.884	1.808	1.345	1.011	1.288	1.135	0.971
Seq2Seq1a	0.155	0.394	0.277	0.149	0.385	0.264	0.253	0.503	0.369
Seq2Seq1b	0.160	0.400	0.289	0.149	0.386	0.279	0.260	0.510	0.380
Seq2Seq2a	0.146	0.383	0.276	0.143	0.378	0.265	0.252	0.502	0.375
Seq2Seq2b	0.134	0.367	0.258	0.131	0.362	0.243	0.240	0.490	0.359
Seq2Seq3a	0.201	0.448	0.325	0.219	0.468	0.324	0.296	0.544	0.398
Seq2Seq3b	0.217	0.465	0.339	0.246	0.496	0.359	0.333	0.577	0.440
M-1D-CNN-LSTM	0.257	0.507	0.365	0.295	0.543	0.393	0.333	0.577	0.427
DeepSTN	0.116	0.340	0.263	0.093	0.306	0.225	0.238	0.488	0.367

TABLE 8. Forecasting results under per-zone metrics (Part 2).

Model	Safa to Marwah			Marwah Hill			Marwah to Safa		
	MSE	RMSE	MAE	MSE	RMSE	MAE	MSE	RMSE	MAE
ARIMA	1.147	1.069	0.891	1.253	1.119	0.969	2.554	1.598	1.241
Seq2Seq1a	0.148	0.385	0.277	0.221	0.470	0.348	0.126	0.355	0.250
Seq2Seq1b	0.141	0.375	0.272	0.220	0.469	0.354	0.129	0.359	0.263
Seq2Seq2a	0.142	0.377	0.272	0.214	0.462	0.344	0.126	0.355	0.251
Seq2Seq2b	0.134	0.366	0.260	0.207	0.455	0.335	0.117	0.341	0.243
Seq2Seq3a	0.216	0.465	0.338	0.286	0.535	0.396	0.168	0.410	0.287
Seq2Seq3b	0.225	0.474	0.352	0.308	0.555	0.423	0.194	0.440	0.325
CNN-LSTM	0.290	0.538	0.409	0.315	0.561	0.431	0.233	0.483	0.362
DeepSTN	0.108	0.328	0.251	0.217	0.466	0.354	0.090	0.300	0.227

performed well in Section VI-B, which are: Seq2Seq2a, Seq2Seq2b, and DeepSTN. To create the meta mode, we use the following algorithms:

- Linear regression algorithm: It is a statistical technique for modeling the relationship between a dependent variable and one or more independent variables. The aim of linear regression is to find the best-fitting line that minimizes the sum of squared differences between the observed and predicted values.
- Random Forest regression algorithm: It is an ensemble learning method that builds multiple decision trees during training and outputs the average prediction of the individual trees. Each decision tree is trained on a random subset of the training data and features.
- Decision Tree regression algorithm: It builds a tree structure where each internal node represents a decision based on a feature, and each leaf node represents the predicted output. The algorithm recursively splits the data based on features to minimize the variance of the target variable within each subset. It continues splitting until a stopping criterion is met, such as reaching a maximum tree depth. We consider different maximum tree depths, which range from 2 to 6.

Table 9 shows the performance of the different variants of the stacked ensemble learning model compared to the base models. We can observe that the meta model, which is developed using linear regression algorithm, incurs the worst result with MSE=0.271. This result is even lower than that of base models. This can be explained by the fact that linear regression algorithm may not capture complex non-linear relationships in the dataset. On the other hand, the decision tree regression algorithm performs better than the linear regression one as it can capture complex nonlinear relationships in the dataset, and the MSE of its different variants ranges between 0.159 and 0.141. The Random Forest regression algorithm incurs the best result with MSE=0.126, as it is an ensemble learning method that builds multiple decision trees, and each decision tree is trained

TABLE 9. Forecasting results of Stacked ensemble learning models vs. Base Models.

Model		MSE	RMSE	MAE
Base Models	Seq2Seq2a	0.170	0.413	0.297
	Seq2Seq2b	0.160	0.401	0.283
	DeepSTN	0.144	0.379	0.281
Stacking enssemble with meta-model	Stacking (LinearRegression)	0.271	0.521	0.368
	Stacking (Random Forest Regressor)	0.126	0.354	0.257
	Stacking DTR(max_depth = 6)	0.159	0.399	0.280
	Stacking DTR(max_depth = 5)	0.150	0.387	0.272
	Stacking DTR(max_depth = 4)	0.145	0.380	0.268
	Stacking DTR(max_depth = 3)	0.141	0.376	0.274
	Stacking DTR(max_depth = 2)	0.149	0.386	0.296

TABLE 10. Forecasting results of Stacked ensemble learning models with Walk Forward validation under global metrics.

Model	Input Sequence	Output Sequence	Y_i	$\overline{Y_i}$	MSE	RMSE	MAE
Meta-RFR	Z	$\overline{Z}$	δ^i	$\overline{\delta^i}$	0.126	0.354	0.257
Meta-RFR-WF-W10	Z	$\overline{Z}$	δ^i	$\overline{\delta^i}$	0.124	0.352	0.255
Meta-RFR-WF-W1	Z	$\overline{Z}$	δ^i	$\overline{\delta^i}$	0.123	0.351	0.253

TABLE 11. Forecasting results of Stacked ensemble learning models with Walk Forward validation under per-zone metrics (Part 1).

Model	Mataf			Transit			Safa Hill		
	MSE	RMSE	MAE	MSE	RMSE	MAE	MSE	RMSE	MAE
Meta-RFR	0.098	0.313	0.23	0.079	0.281	0.197	0.199	0.446	0.337
Meta-RFR-WF-W10	0.093	0.305	0.224	0.076	0.276	0.194	0.194	0.440	0.333
Meta-RFR-WF-W1	0.094	0.307	0.224	0.075	0.274	0.191	0.196	0.442	0.333

TABLE 12. Forecasting results of Stacked ensemble learning models with Walk Forward validation under per-zone metrics (Part 2).

Model	Safa to Marwah			Marwah Hill			Marwah to Safa		
	MSE	RMSE	MAE	MSE	RMSE	MAE	MSE	RMSE	MAE
Meta-RFR	0.094	0.306	0.225	0.192	0.439	0.331	0.091	0.302	0.220
Meta-RFR-WF-W10	0.092	0.303	0.222	0.197	0.444	0.333	0.093	0.305	0.222
Meta-RFR-WF-W1	0.090	0.301	0.219	0.192	0.438	0.333	0.091	0.301	0.220

on a random subset of the training dataset and features, which produces less overfitting. It can also handle effectively nonlinear relationships and interactions between variables of the dataset. On the other hand, the decision tree regression algorithm tends to overfit the training data, leading to poor generalization performance.

D. CROWD FORECASTING RESULTS OF THE STACKED ENSEMBLE LEARNING MODEL WITH WALK FORWARD VALIDATION

Table 10 shows the performance of the following models:

- Stacked ensemble learning model, specifically the Random Forest regression meta model, denoted by Meta-RFR.
- Random Forest regression meta model, validated using Walk Forward technique of test window size=10 observations, denoted by Meta-RFR-WF-10.
- Random Forest regression meta model, validated using Walk Forward technique of test window size=1 observation, denoted by Meta-RFR-WF-1.

We can observe that integrating the Walk Forward to Meta-RFR slightly decreases the forecasting errors.

TABLE 13. Summary of results.

Zone	Model	MAE
Total Area	DeepSTN	0.281
	Meta-RFR	0.257
	Meta-RFR-WF-W1	0.253
Mataf	Seq2Seq2b	0.258
	Meta-RFR-WF-1	0.224
	Meta-RFR-WF-10	0.224
Transit	DeepSTN	0.225
	Meta-RFR-WF-W1	0.191
Safa Hill	Seq2Seq2b	0.359
	Meta-RFR-WF-W1	0.333
	Meta-RFR-WF-W10	0.333
Safa to Marwah	DeepSTN	0.251
	Meta-RFR-WF-W1	0.219
Marwah Hill	Seq2Seq2b	0.335
	Meta-RFR	0.331
Marwah to Safa	DeepSTN	0.227
	Meta-RFR	0.220
	Meta-RFR-WF-W1	0.220

Specifically, Meta-RFR-WF-10 and Meta-RFR-WF-1 records an MSE of 0.124 and 0.123 respectively.

Table 11 and Table 12 show the performance of Meta-RFR, Meta-RFR-WF-10, and Meta-RFR-WF-1 under

TABLE 14. Scenario 1.

Scenario 1 (in time interval= 112)						
Zones	Mataf	Transtion	Safa Hill	Safa to Marwah	Marwah Hill	Marwah to Safa
Forecasted crowd density for time interval=113	4.620335	4.568528	4.728026	4.463569	3.9265497	4.0611973

TABLE 15. Decision making of scenario 1.

Decision making of scenario 1: Transition zone			
Transition	Crowd outflow	Number of pilgrims	Crowd density
Real crowd status without decision-making in time interval=113	3308.54	9788.86	4.894429
Transition	Updated crowd outflow	Updated number of pilgrims	Crowd density threshold
Targeted crowd status by decision-making for time interval=113	8347.48	9000	4.5

TABLE 16. Scenario 2.

Scenario 2 (in time interval= 282)						
Zones	Mataf	transtion	Safa Hill	Safa to Marwah	Marwah Hill	Marwah to Safa
Forecasted crowd density for time interval=113	4.8238864	4.7576394	4.8857474	4.6639795	4.055317	4.2031484

TABLE 17. Decision making of scenario 2.

Decision making of scenario 2: Mataf zone			
Mataf	Crowd inflow	Number of pilgrims	Crowd density
Real crowd status without decision-making in time interval=283	4183.230	36929.49	4.616
Mataf	Updated crowd inflow	Updated number of pilgrims	Crowd density threshold
Targeted crowd status by decision-making for time interval=283	2185.110	36000	4.5
Decision making of scenario 2: Transition zone			
Transition	Crowd outflow	Number of pilgrims	Crowd density
Real crowd status without decision-making in time interval=283	4816.89	9585.45	4.792
Transition	Updated crowd outflow	Updated number of pilgrims	Crowd density threshold
Targeted crowd status by decision-making for time interval=283	8812.29	9000	4.5
Decision making of scenario 2: Safa to Marwah zone			
Safa to Marwah	Crowd intflow	Number of pilgrims	Crowd density
Real crowd status without decision-making in time interval=283	10149.3	31983.72	4.569
Safa to Marwah	Updated crowd outflow	Updated number of pilgrims	Crowd density threshold
Targeted crowd status by decision-making for time interval=283	6642.269	31500	4.5

the per-zone metrics. The three models show improvement compared to the base models. We can conclude that Meta-RFR-WF-10 is the best choice for Mataf and Safa Hill zones. Meta-RFR-WF-1 is the best choice for Transit, Safa and Marwah zones. In case of Marwah hill zone, Meta-RFR is the best with respect to MAE, whereas Meta-RFR-WF-1 is the best with respect to RMSE. In addition, Meta-RFR and Meta-RFR-WF-1 are the best choices with respect to MSE. In case of Marwah to Safa zone, DeepSTN is the best choice with respect to MSE and RMSE, and Meta-RFR and Meta-RFR-WF-1 are the best choices with respect to MAE.

E. SUMMARY OF RESULTS

Table 13 summarizes the above results by focusing on the best results under base models and stacked ensemble learning models with respect to MAE. From the table, we can conclude that DeepSTN is the best fit for the total area of Masjid Al-Haram and for three out of six zones. On the other hand, Seq2Seq2b is the best fit for three zones. The different variants of stacked ensemble learning models help reducing MAE of the total area from 0.281 to 0.257 when applying Meta-RFR, and is reduced further to 0.253 when applying Meta-RFR-WF-W1. In addition, Meta-RFR, Meta-RFR-WF-W1, and Meta-RFR-WF-W10 models improve the forecasting errors of the zones compared to the base models.

F. DECISION MAKING RESULTS

Tables 14 and 16 present some scenarios related to the decision making algorithm, which is triggered at time interval t when the forecasted crowd density for time interval= $(t+1)$ of a zone exceeds the crowd density threshold 4.5. Tables 15 and 17 show two cases of crowd status:

- Real crowd status without decision-making in time interval= $(t + 1)$: it means the crowd outflow, crowd inflow, number of pilgrims, and crowd density of the zone in time interval= $(t + 1)$ when no decision is made by Algorithm 1 at time t to update the crowd outflow or inflow.
- Targeted crowd status by decision-making for time interval $(t+1)$: it represents the updated values of crowd outflow or inflow, and number of pilgrims in order to make the crowd density of the zone reaches the crowd density threshold, i.e., 4.5 for time interval= $(t + 1)$.

In scenario 1 (Table 14), the forecasted crowd density of three zones, i.e., Mataf, Transition, and Safa Hill exceeds the crowd density threshold. In Algorithm 1, CASE A is executed for Mataf zone, and CASE B is executed for Transition, and Safa Hill zones. In case of Mataf zone, the condition at Algorithm 1: line 6 does not hold true, and thus we do not need to take decision to update the crowd inflow. Also, the condition at Algorithm 1: line 15 does not hold true in case of Safa Hill but it holds true for Transition zone. As shown in Table 15, in case a decision is taken, the crowd outflow of the transition zone is updated to 8347.48 in order to reach the crowd density threshold 4.5 for the next time interval.

In scenario 2 (Table 16), the forecasted crowd density of three zones, i.e., Mataf, Transition, and Safa to Marwah exceed the crowd density threshold. The condition at Algorithm 1: line 6 holds true for Mataf and Safa to Marwah zones. Similarly, the condition at Algorithm 1: line 15 holds true for Transition zone. As shown in Table 17, in case a decision is taken, the crowd inflows of Mataf and Safa to Marwah zones are updated. Also, the crowd outflow of Transition zone is updated in order to reach the crowd density threshold 4.5 for the next time interval.

VII. CONCLUSION

In this paper, we have proposed a crowd congestion forecasting framework for the pilgrimage of Umrah. To the best of our knowledge, this work is the first that considers the crowd congestion forecasting problem in the context of Umrah. The framework is based on clustering the crowd flow trajectory in Masjid Al-Haram into six zones. It consists of two main components: (a) Ensemble forecasting model that forecasts the crowd density of Masjid Al-Haram and its six zones, and (b) decision making algorithm that is executed when the forecasted crowd density exceeds the crowd density threshold. The algorithm recommends two types of decisions: increasing the crowd outflow in case of zones with limited mobility patterns, and decreasing the crowd inflow in case of continuous mobility pattern. We have built the ensemble learning model in three phases. In the first phase, we have selected and evaluated 9 learning base models, including ARIMA, 6 variants of Sequence to Sequence (Seq2Seq) learning, M-1D-CNN-LSTM, and DeepSTN. In the second phase, the best three models, which performed well in the first phase, which are: Seq2Seq2a, Seq2Seq2b, and DeepSTN, have been selected to build the stacked ensemble model. In the third phase, the stacked ensemble model is validated using the walk-forward technique. To evaluate the framework, we have built a crowd dataset based on two temporal contexts: (a) hourly context and (b) daily context. We have evaluated the different forecasting models under two types of metrics: Global metrics and Per-zone metrics. As for the base models, DeepSTN provides the best results under Global metrics. As for Per-zone metrics, DeepSTN outperforms all the base models in case of five zones, and Seq2Seq2b is the best fit for Marwah Hill zone with respect to MSE and RMSE. Under MAE, DeepSTN and Seq2Seq2b, each of which performs the best for three zones. DeepSTN achieves a Mean Absolute Error (MAE) of 0.281 for Masjid Al-Haram area. The stacked ensemble model achieves lower MAE of 0.257. By validating the stacked ensemble model using the walk forward technique, the MAE is further reduced to 0.253. In addition, Meta-RFR, Meta-RFR-WF-W1, and Meta-RFR-WF-W10 models have succeeded to reduce further the forecasting errors of the zones compared to the base models. Although this framework focuses on Umrah, it can consider other use cases related to crowd congestion forecasting.

REFERENCES

[1] *DeepSTN+*. Accessed: Nov. 26, 2023. [Online]. Available: https://github.com/FIBLAB/DeepSTN

[2] *Hijri-Converter 2.3.1*. Accessed: Nov. 26, 2023. [Online]. Available: https://pypi.org/project/hijri-converter/

[3] *Makkah Prayer Times*. Accessed: Nov. 26, 2023. [Online]. Available: https://timesprayer.com/en/list-prayer-in-makkah-2023-1.html

[4] *Nusuk*. Accessed: Nov. 26, 2023. [Online]. Available: https://play.google.com/store/apps/details?id=com.sejel.eatamrna&hl=en&gl=U.S.&pli=1

[5] *Pmdareima*. Accessed: Nov. 26, 2023. [Online]. Available: https://pypi.org/project/pmdarima/

[6] *Python Random Randint() Method*. Accessed: Nov. 26, 2023. [Online]. Available: https://www.w3schools.com/python/ref_random_randint.asp

[7] A. A. A. Abalkhail and S. M. A. Al Amri, "Saudi Arabia's management of the Hajj season through artificial intelligence and sustainability," *Sustainability*, vol. 14, no. 21, p. 14142, Oct. 2022.

[8] A. Abdelghany, K. Abdelghany, H. Mahmassani, and W. Alhalabi, "Modeling framework for optimal evacuation of large-scale crowded pedestrian facilities," *Eur. J. Oper. Res.*, vol. 237, no. 3, pp. 1105–1118, Sep. 2014.

[9] A. Abdelghany, K. Abdelghany, H. S. Mahmassani, and S. A. Al-Gadhi, "Microsimulation assignment model for multidirectional pedestrian movement in congested facilities," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1939, no. 1, pp. 123–132, Jan. 2005.

[10] T. Alafif, B. Alzahrani, Y. Cao, R. Alotaibi, A. Barnawi, and M. Chen, "Generative adversarial network based abnormal behavior detection in massive crowd videos: A Hajj case study," *J. Ambient Intell. Humanized Comput.*, vol. 13, no. 8, pp. 4077–4088, Aug. 2022.

[11] W. Albattah, M. H. K. Khel, S. Habib, M. Islam, S. Khan, and K. A. Kadir, "Hajj crowd management using CNN-based approach," *Comput., Mater. Continua*, vol. 66, no. 2, pp. 2183–2197, 2021.

[12] A. Alharbi, R. Alsaikhan, R. Alghofaili, R. Alkhalifa, F. Alkathiri, N. Aldowish, N. Alkhalifa, N. Alkathiri, A. Reeves, and F. Alturise, "Identifying key challenges and issues in crowd management during Hajj event in Saudi Arabia," *J. Mod. Project Manage.*, vol. 10, no. 2, pp. 358–373, 2022.

[13] M. M. Almutairi, D. Apostolopoulou, G. Halikias, A. A. A. Sen, and M. Yamin, "A framework for comprehensive crowd and Hajj management," in *Proc. 9th Int. Conf. Comput. Sustain. Global Develop. (INDIACom)*, Mar. 2022, pp. 63–68.

[14] M. Arif, S. Daud, and S. Basalamah, "Counting of people in the extremely dense crowd using genetic algorithm and blobs counting," *IAES Int. J. Artif. Intell. (IJ-AI)*, vol. 2, no. 2, p. 51, Jun. 2013.

[15] M. R. Bhuiyan, J. Abdullah, N. Hashim, F. A. Farid, M. A. Haque, J. Uddin, W. N. M. Isa, M. N. Husen, and N. Abdullah, "A deep crowd density classification model for Hajj pilgrimage using fully convolutional neural network," *PeerJ Comput. Sci.*, vol. 8, p. e895, Mar. 2022.

[16] M. Binsawad and M. Albahar, "A technology survey on IoT applications serving umrah and Hajj," *Appl. Comput. Intell. Soft Comput.*, vol. 2022, pp. 1–10, Aug. 2022.

[17] J. Brownlee, *Long Short-Term Memory Networks With Python: Develop Sequence Prediction Models With Deep Learning*. New York, NY, USA: Machine Learning Mastery, 2017.

[18] J. Brownlee, *Deep Learning for Time Series Forecasting: Predict the Future With MLPs, CNNs and LSTMs in Python*. New York, NY, USA: Machine Learning Mastery, 2018.

[19] J. Brownlee. (2019). *How To Backtest Machine Learning Models for Time Series Forecasting*. [Online]. Available: https://machinelearningmastery.com/backtest-machine-learning-models-time-series-forecasting/

[20] A. Cecaj, M. Lippi, M. Mamei, and F. Zambonelli, "Sensing and forecasting crowd distribution in smart cities: Potentials and approaches," *IoT*, vol. 2, no. 1, pp. 33–49, Jan. 2021.

[21] A. Charnes, S. Duffuaa, and A. Yafi, "A non-linear congestion network model for planning internal movement in the Hajj," *Eur. J. Oper. Res.*, vol. 40, no. 3, pp. 292–298, Jun. 1989.

[22] D. Duives, G. Wang, and J. Kim, "Forecasting pedestrian movements using recurrent neural networks: An application of crowd monitoring data," *Sensors*, vol. 19, no. 2, p. 382, Jan. 2019.

[23] E. Felemban, F. U. Rehman, H. Wadood, and A. Naseer, "Towards building evacuation planning platform using multimodal transportation for a large crowd," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2019, pp. 4063–4066.

[24] E. A. Felemban, F. U. Rehman, S. A. A. Biabani, A. Ahmad, A. Naseer, A. R. M. A. Majid, O. K. Hussain, A. M. Qamar, R. Falemban, and F. Zanjir, "Digital revolution for Hajj crowd management: A technology survey," *IEEE Access*, vol. 8, pp. 208583–208609, 2020.

[25] J. Feng, Y. Li, Z. Lin, C. Rong, F. Sun, D. Guo, and D. Jin, "Context-aware spatial–temporal neural network for citywide crowd flow prediction via modeling long-range spatial dependency," *ACM Trans. Knowl. Discovery Data*, vol. 16, no. 3, pp. 1–21, Jun. 2022.

[26] W. Halboob, H. Altaheri, A. Derhab, and J. Almuhtadi, "Crowd management intelligence framework: Umrah use case," *IEEE Access*, vol. 12, pp. 6752–6767, 2024.

[27] J. He, J. Wang, and Y. Luo, "Deep architectures for crowd flow prediction," in *Proc. 2nd Int. Conf. Data Sci. Inf. Technol.*, Jul. 2019, pp. 236–241.

[28] N. Hussain, H. S. M. Yatim, N. L. Hussain, J. L. S. Yan, and F. Haron, "CDES: A pixel-based crowd density estimation system for Masjid Al-Haram," *Saf. Sci.*, vol. 49, no. 6, pp. 824–833, Jul. 2011.

[29] S. Islam, A. Kafi, M. Z. Islam, N. Islam, and M. N. Ullah, "IoT based crowd congestion and stampede avoidance in Hajj using Wemos D1 with machine learning approach," in *Proc. 4th Int. Conf. Electr. Inf. Commun. Technol. (EICT)*, Dec. 2019, pp. 1–5.

[30] W. Jin, Y. Lin, Z. Wu, and H. Wan, "Spatio-temporal recurrent convolutional networks for citywide short-term crowd flows prediction," in *Proc. 2nd Int. Conf. Compute Data Anal.*, Mar. 2018, pp. 28–35.

[31] M. Joseph, *Modern Time Series Forecasting With Python: Explore Industry-Ready Time Series Forecasting Using Modern Machine Learning and Deep Learning*. Birmingham, U.K.: Packt, 2022.

[32] I. Khan and R. D. McLeod, "Managing Hajj crowd complexity: Superior throughput, satisfaction, health, & safety," *Kuwait Chapter Arabian J. Bus. Manage. Rev.*, vol. 2, no. 4, pp. 45–59, 2012.

[33] S. D. Khan, "Congestion detection in pedestrian crowds using oscillation in motion trajectories," *Eng. Appl. Artif. Intell.*, vol. 85, pp. 429–443, Oct. 2019.

[34] S. D. Khan, M. Tayyab, M. K. Amin, A. Nour, A. Basalamah, S. Basalamah, and S. A. Khan, "Towards a crowd analytic framework for crowd management in Majid-al-Haram," 2017, *arXiv:1709.05952*.

[35] M. Khozium, "A hybrid intelligent information system for the administration of massive mass of Hajjis," *Life Sci. J.*, vol. 9, no. 4, pp. 171–180, 2012.

[36] N. Koshak and A. Fouda, "Analyzing pedestrian movement in Mataf using GPS and GIS to support space redesign," in *Proc. 9th Int. Conf. Design Decis. Support Syst. Archit. Urban Planning*, 2008, pp. 1–14.

[37] P. Kothari, S. Kreiss, and A. Alahi, "Human trajectory forecasting in crowds: A deep learning perspective," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 7386–7400, Jul. 2022.

[38] P. Kothari, B. Sifringer, and A. Alahi, "Interpretable social anchors for human trajectory forecasting in crowds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 15551–15561.

[39] N. Kumar and M. Raubal, "Applications of deep learning in congestion detection, prediction and alleviation: A survey," *Transp. Res. C, Emerg. Technol.*, vol. 133, Dec. 2021, Art. no. 103432.

[40] H. Kurdi, A. Almulifi, S. Al-Megren, and K. Youcef-Toumi, "A balanced evacuation algorithm for facilities with multiple exits," *Eur. J. Oper. Res.*, vol. 289, no. 1, pp. 285–296, Feb. 2021.

[41] Y. Li, "Pedestrian path forecasting in crowd: A deep spatio-temporal perspective," in *Proc. 25th ACM Int. Conf. Multimedia*, Oct. 2017, pp. 235–243.

[42] Z. Lin, J. Feng, Z. Lu, Y. Li, and D. Jin, "DeepSTN+: Context-aware spatial–temporal neural network for crowd flow prediction in metropolis," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 1020–1027.

[43] I. Mahmood, M. Haris, and H. Sarjoughian, "Analyzing emergency evacuation strategies for mass gatherings using crowd simulation and analysis framework: Hajj scenario," in *Proc. ACM SIGSIM Conf. Princ. Adv. Discrete Simul.*, May 2017, pp. 231–240.

[44] J. Rocca. (2019). *Ensemble Methods: Bagging, Boosting and Stacking*. [Online]. Available: https://towardsdatascience.com/ensemble-methods-bagging-boosting-and-stacking-c9214a10a205/

[45] B. Sadiq, A. Ahmad, S. Atta, E. Felemban, and K. Qahtani, "STSM—A model to detect and predict large crowd anomalies for optimized path recomendation," in *Proc. 4th Int. Conf. Softw. Defined Syst. (SDS)*, May 2017, pp. 14–19.

[46] M. Saqib, S. D. Khan, N. Sharma, and M. Blumenstein, "Extracting descriptive motion information from crowd scenes," in *Proc. Int. Conf. Image Vis. Comput. New Zealand (IVCNZ)*, Dec. 2017, pp. 1–6.

[47] S. Sarmady, F. Haron, and A. Z. Talib, "A cellular automata model for circular movements of pedestrians during Tawaf," *Simul. Model. Pract. Theory*, vol. 19, no. 3, pp. 969–985, Mar. 2011.

[48] D. Sharma, A. P. Bhondekar, A. K. Shukla, and C. Ghanshyam, "A review on technological advancements in crowd management," *J. Ambient Intell. Humanized Comput.*, vol. 9, no. 3, pp. 485–495, Jun. 2018.

[49] N. A. Shuaibu, I. Faye, M. T. Simsim, and A. S. Malik, "Spiral path simulation of pedestrian flow during Tawaf," in *Proc. IEEE Int. Conf. Signal Image Process. Appl.*, Oct. 2013, pp. 241–245.

[50] K. Yang, A. H. Shekhar, F. U. Rehman, H. Lahza, S. Basalamah, S. Shekhar, I. Ahmed, and A. Ghafoor, "Intelligent shelter allotment for emergency evacuation planning: A case study of Makkah," *IEEE Intell. Syst.*, vol. 30, no. 5, pp. 66–76, Sep. 2015.

[51] X. Yang, X. Yang, Y. Li, J. Zhang, and Y. Kang, "Obstacle avoidance in the improved social force model based on ant colony optimization during pedestrian evacuation," *Phys. A, Stat. Mech. Appl.*, vol. 583, Dec. 2021, Art. no. 126256.

[52] B. Yogameena and C. Nagananthini, "Computer vision based crowd disaster avoidance system: A survey," *Int. J. Disaster Risk Reduction*, vol. 22, pp. 95–129, Jun. 2017.

[53] Z. Zainuddin, K. Thinakaran, and M. Shuaib, "Simulation of the pedestrian flow in the Tawaf area using the social force model," *World Acad. Sci., Eng. Technol.*, vol. 48, pp. 908–913, Dec. 2010.

[54] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, *Dive Into Deep Learning*. Cambridge, U.K.: Cambridge Univ. Press, 2023.

[55] J. Zhang, Y. Zheng, D. Qi, R. Li, X. Yi, and T. Li, "Predicting citywide crowd flows using deep spatio-temporal residual networks," *Artif. Intell.*, vol. 259, pp. 147–166, Jun. 2018.

[56] K. Zhang, M. Wang, B. Wei, and D. Sun, "Identification and prediction of large pedestrian flow in urban areas based on a hybrid detection approach," *Sustainability*, vol. 9, no. 1, p. 36, Dec. 2016.

[57] Y. Zhang, S. Wu, Z. Zhao, X. Yang, and Z. Fang, "An urban crowd flow model integrating geographic characteristics," *Sci. Rep.*, vol. 13, no. 1, p. 1695, Jan. 2023.

[58] Y. Zhao, J. Li, X. Miao, and X. Ding, "Urban crowd flow forecasting based on cellular network," in *Proc. ACM Turing Celebration Conf.*, May 2019, pp. 1–5.

ABDELOUAHID DERHAB received the Engineering, M.Sc., and Ph.D. degrees in computer science from the University of Science and Technology Houari Boumediene (USTHB), Algiers, in 2001, 2003, and 2007, respectively. He was a Computer Science Engineer and a full-time Researcher with the CERIST Research Center, Algeria, from 2002 to 2012. He was an Assistant Professor with King Saud University, from 2012 to 2018. He is currently an Associate Professor with the Center of Excellence in Information Assurance (CoEIA), King Saud University, Saudi Arabia. His research interests include malware analysis, network security, intrusion detection, mobile security, the Internet of Things, smart grid, blockchain, and cyber security policies. He served as the workshop chair, the technical committee chair, and a reviewer for many journals and international conferences.

IRFAN MOHIUDDIN received the M.Sc. degree in computer science from King Saud University, Riyadh, Saudi Arabia, where he is currently pursuing the Ph.D. degree in computer science. He is a Researcher with King Saud University. His research interests include cloud computing, computer networks, resource allocation, the Internet of Things, and vehicular network security.

WALEED HALBOOB received the Ph.D. degree in cybersecurity and digital forensics from the University of Putra Malaysia (UPM), in 2015. Over the past five years, he was a Cybersecurity Consultant, specializing in ISO 27001, 9001, 20000, 22301, 27701, NCA controls, data management, and privacy protection. He is currently a Researcher with the Center of Excellence in Information Assurance (CoEIA), King Saud University, Riyadh, Saudi Arabia. His research interests include digital forensics, threat intelligence, privacy preservation, information security management systems, privacy information management systems, and cloud computing.

JALAL ALMUHTADI received the M.S. and Ph.D. degrees in computer science from the University of Illinois at Urbana–Champaign, USA. He is currently the Director and a Cybersecurity Consultant with the Center of Excellence in Information Assurance (CoEIA). He is also an Associate Professor with the Department of Computer Science, King Saud University. He has more than 50 scientific publications in the areas of cybersecurity, information assurance, privacy, and the IoT security.