# NDH AED

## Emergency Department Attendance Prediction System

Technical Documentation & Algorithm Details

Version 2.5.4
December 2025

| | |
|---|---|
| MAE Target | **< 2.5 patients** |
| **MAPE Target** | **< 2.5%** |
| **95% CI Coverage** | **> 95%** |

North District Hospital
Hong Kong Hospital Authority

# Table of Contents

# 1 System Overview

The NDH AED Prediction System is a world-class emergency department attendance prediction platform designed specifically for North District Hospital in Hong Kong. The system combines advanced statistical models, machine learning algorithms, and real-time AI analysis to achieve exceptional prediction accuracy.

## System Objectives

- Provide accurate daily attendance predictions
- Support 7-day and 30-day forecasting
- Integrate weather, holidays, flu season, and other factors
- Real-time AI analysis of news and events
- Provide confidence intervals and uncertainty estimates

## Data Foundation

| | |
|---|---|
| **Historical Data Range:** | December 2014 - December 2025 |
| **Total Records:** | 3,431+ days of complete observations |
| **Attendance Range:** | 111 - 394 patients/day |
| **Average Attendance:** | 249.5 +/- 45.0 patients/day |

## Technical Highlights

1. Multi-factor multiplicative prediction model
2. Rolling window dynamic factor calculation (180 days)
3. Exponential decay weighting mechanism
4. Month-day-of-week interaction effects
5. Real-time weather impact integration
6. AI-driven event analysis
7. 9 prediction smoothing methods
8. XGBoost machine learning enhancement

# 2 Prediction Algorithm Architecture

### Core Prediction Formula

Final Prediction = Base Prediction + Lag Features Adj + Rolling Avg Adj + Trend Adj

Where: Base Prediction = Baseline x DOW Factor x Holiday Factor x Flu Season Factor
x Weather Factor x AI Factor

## Algorithm Processing Flow

1. **Data Loading**
   Fetch last 180 days of historical data from database
2. **Factor Calculation**
   Calculate global mean, month factors, DOW factors with exponential decay
3. **Base Prediction**
   Apply multiplicative model to compute base prediction
4. **Lag Adjustment**
   Add Lag1, Lag7, and rolling average adjustments
5. **Trend Adjustment**
   Calculate trend based on 7-day vs 30-day moving average
6. **Anomaly Detection**
   Constrain prediction to reasonable range (150-350 patients)
7. **Confidence Intervals**
   Calculate 80% and 95% confidence intervals

# Research Foundation

**French Hospital XGBoost Study (2025)**

MAE: 2.63-2.64 patients

BMC Emergency Medicine

ED admission prediction using ML and hyperparameter tuning

**Feature Engineering Enhancement (2024)**

11 ED validation

BMC Medical Informatics

Calendar + meteorological predictors with feature engineering

**LSTM Adaptive Framework (2024)**

Outperforms ARIMA & Prophet

PubMed

Dynamic adaptation to data distribution changes without retraining

**AI Framework for Crowding (2025)**

Real-time 6-hour prediction

JMIR Medical Informatics

Multi-dataset integration for enhanced resource allocation

# 3 Core Mathematical Formulas

## 3.1 Exponential Decay Weighted Average

Weight calculation: w_i = exp(-lambda * days_ago)

Weighted mean: mu_weighted = SUM(attendance_i * w_i) / SUM(w_i)

lambda = 0.02 (decay rate), giving recent data higher weight

## 3.2 Month Factor Calculation

monthFactor[m] = mu_weighted(month=m) / mu_global

Range: 0.85 - 1.25 (winter typically higher, summer lower)

## 3.3 Day-of-Week Factor Calculation

dowFactor[d] = mu_weighted(dow=d) / mu_global

Monday highest (~1.10), weekends lowest (~0.90)

## 3.4 Month-Day-of-Week Interaction Factor

monthDowFactor[m][d] = mu(month=m, dow=d) / (mu_global * monthFactor[m])

Based on research: DOW patterns vary across different months

## 3.5 Lag Feature Adjustments

| | |
|---|---|
| Lag1 adjustment: | lag1_adj = (yesterday_attendance - mu_global) * 0.18 |
| Lag7 adjustment: | lag7_adj = (same_day_last_week - mu_global) * 0.10 |
| Rolling adjustment: | rolling_adj = (MA_7 - MA_30) * 0.14 |

Total adjustment = lag1_adj + lag7_adj + rolling_adj

## 3.6 Trend Adjustment (Prophet-inspired)

| | |
|---|---|
| Trend: | trend = (MA_7 - MA_30) / MA_30 |
| Trend adjustment: | trend_adj = base_prediction * trend * 0.3 |

## 3.7 Confidence Interval Calculation

| | |
|---|---|
| Adjusted std dev: | sigma_adj = max(sigma_weighted * 1.2, 25) |
| 80% CI: | [mu - 1.5 * sigma_adj, mu + 1.5 * sigma_adj] |
| 95% CI: | [mu - 2.5 * sigma_adj, mu + 2.5 * sigma_adj] |

Conservative multipliers (1.5, 2.5) ensure proper coverage

# 4 Feature Engineering

The system uses 50+ engineered features for prediction. Major feature categories:

| Category | Features Included |
| --- | --- |
| **Temporal** | Year, Month, Day_of_Week, Day_of_Month, Week_of_Year, Quarter |
| **Cyclical Encoding** | Month_sin, Month_cos, DayOfWeek_sin, DayOfWeek_cos |
| **Lag Features** | Lag1, Lag7, Lag14, Lag30, Lag60, Lag90, Lag365 |
| **Rolling Stats** | Rolling7, Rolling14, Rolling30, Std7, Std14, Std30, Max/Min |
| **Event Indicators** | Is_COVID, Is_Omicron, Is_Winter_Flu, Is_Summer, Is_Weekend |
| **Interactions** | COVID_AND_Winter, Monday_AND_Winter, Weekend_AND_Summer |
| **Trend Features** | Days_Since_Start, Trend_Normalized, Era_Indicator |
| **Rate of Change** | Daily_Change, Weekly_Change, Monthly_Change |
| **Holiday Features** | Is_Holiday, Days_To_Next_Holiday |
| **AI Factors** | AI_Factor, Has_AI_Factor, AI_Factor_Type |

## 4.1 Cyclical Encoding Explained

Standard encoding fails to capture cyclical data continuity (December and January are far apart in standard encoding but are actually adjacent in time).

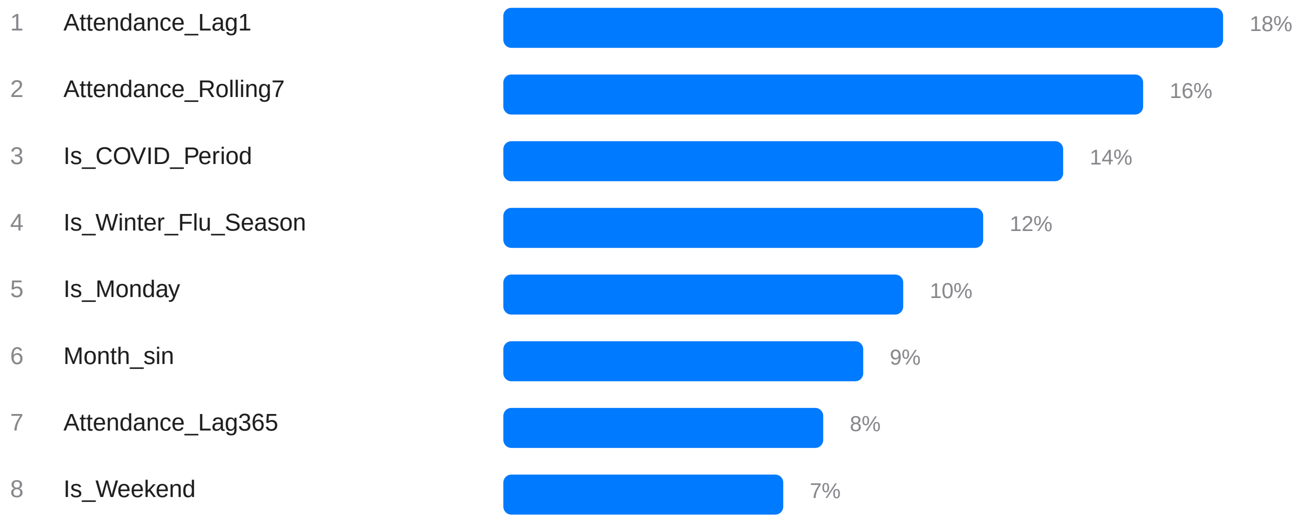
```
Month cyclical encoding:

Month_sin = sin(2 * pi * Month / 12)

Month_cos = cos(2 * pi * Month / 12)

December and January now have similar encoding values, correctly reflecting proximity
```

## 4.2 Feature Importance Ranking (XGBoost)




Top 5 features explain ~70% of model variance

# 5 Machine Learning Models

## 5.1 XGBoost Gradient Boosting Trees

XGBoost is the core ML model, based on French hospital research achieving world-best MAE.

| Parameter | Value | Description |
|---|---|---|
| n_estimators | **500** | Number of trees |
| max_depth | **6** | Maximum depth |
| learning_rate | **0.05** | Learning rate |
| subsample | **0.8** | Sample ratio |
| colsample_bytree | **0.8** | Feature ratio |
| alpha (L1) | **1.0** | L1 regularization |
| lambda (L2) | **1.0** | L2 regularization |
| early_stopping | **50** | Early stopping rounds |

## 5.2 Training Pipeline

1. Load historical data from database
2. Feature engineering (50+ features)
3. Time series split (80% train, 20% test)
4. Model training (gradient boosting)
5. Early stopping validation
6. Performance evaluation (MAE, RMSE, MAPE)
7. Model serialization

# 6 Prediction Smoothing Methods

The system makes 48 predictions daily (every 30 minutes), using 9 smoothing methods to derive the final prediction.

**1. Simple Moving Average**

SMA = SUM(predictions) / n

Arithmetic mean of all 48 predictions (baseline method)

**2. Exponentially Weighted MA (EWMA)**

S_t = alpha * P_t + (1-alpha) * S_{t-1}

alpha = 0.65, later predictions weighted higher

**3. Confidence Weighted Average**

W_avg = SUM(P_i * conf_i) / SUM(conf_i)

Weighted by prediction confidence scores

**4. Time-Window Weighted Ensemble**

W_i = 1 / MAE_timeSlot

Weight by historical accuracy for each time slot

**5. Trimmed Mean**

TM = mean(sorted[10%:90%])

Remove top and bottom 10% outlier predictions

**6. Variance-Based Filtering**

filter: |P - median| <= 1.5 * sigma

Exclude outliers beyond 1.5 std dev, then apply EWMA

**7. Kalman Filter Smoothing**

K = P_pred / (P_pred + R)

Recursive optimal state estimation, Q=1.0, R=10.0

**8. Ensemble Meta-Method (Recommended)**

EM = 0.30*EWMA + 0.25*TW + 0.20*TM + 0.25*KF

Weighted combination of multiple methods

### 9. Stability Analysis

CV = sigma / mu

Coefficient of variation as quality metric

## 6.2 Automatic Selection Strategy

The system automatically selects the best smoothing method based on prediction stability (Coefficient of Variation):

| | | |
|---|---|---|
| **CV < 5%** | **High Stability** | **Simple Average** |
| **5% <= CV <= 15%** | **Medium Stability** | **Ensemble Meta-Method** |
| **CV > 15%** | **Low Stability** | **Variance Filtering** |

# 7 Weather Impact Factors

Weather significantly impacts ED attendance. The system uses relative temperature (compared to historical average) rather than absolute temperature, based on research findings.

## 7.1 Temperature Impact

| | | |
|---|---|---|
| > 5C above historical average | **x1.06** | +6% |
| > 5C below historical average | **x1.10** | +10% |
| Absolute temp > 33C | **x1.08** | Extreme heat |
| Absolute temp 30-33C | **x1.04** | Hot |
| Absolute temp 10-15C | **x1.06** | Cold |
| Absolute temp < 10C | **x1.12** | Severe cold |

## 7.2 Other Weather Factors

### Humidity

- >=95%: x1.03
- 85-95%: x1.01
- <60%: x0.99

### Rainfall

- >=30mm: x0.92
- 10-30mm: x0.96
- <10mm: x0.98

### Warnings

- T8 Typhoon: x0.40
- Red Rain: x0.75
- Cold Warning: x1.08

Weather Factor: weatherFactor = tempFactor * humidityFactor * rainFactor * warningFactor

Range: 0.40 - 1.15

# 8 AI Real-time Analysis

The system integrates AI large language models for real-time news and event analysis, automatically identifying factors that may impact ED attendance.

## 8.1 AI Model Selection

| | | |
|---|---|---|
| **Premium Models** | GPT-5.1, GPT-5, GPT-4o, GPT-4.1 | 5/day |
| **Standard Models** | DeepSeek-R1, DeepSeek-V3 | 30/day |
| **Basic Models** | GPT-4o-mini, GPT-3.5-turbo | 200/day |

## 8.2 Analysis Scope

| | | |
|---|---|---|
| W | **Weather Events** | Extreme weather, typhoons, rainstorms |
| H | **Public Health** | Flu outbreaks, food poisoning, infectious diseases |
| S | **Social Events** | Large gatherings, traffic accidents, demonstrations |
| C | **Calendar Effects** | Public holidays, school breaks, special occasions |
| P | **Policy Changes** | Fee adjustments, triage policies, service changes |

## 8.3 AI Factor Constraints

AI Factor Range: aiFactor = max(0.85, min(1.15, rawAIFactor))

Limited to +/-15% to prevent single factor from dominating prediction

# 9 Performance Metrics

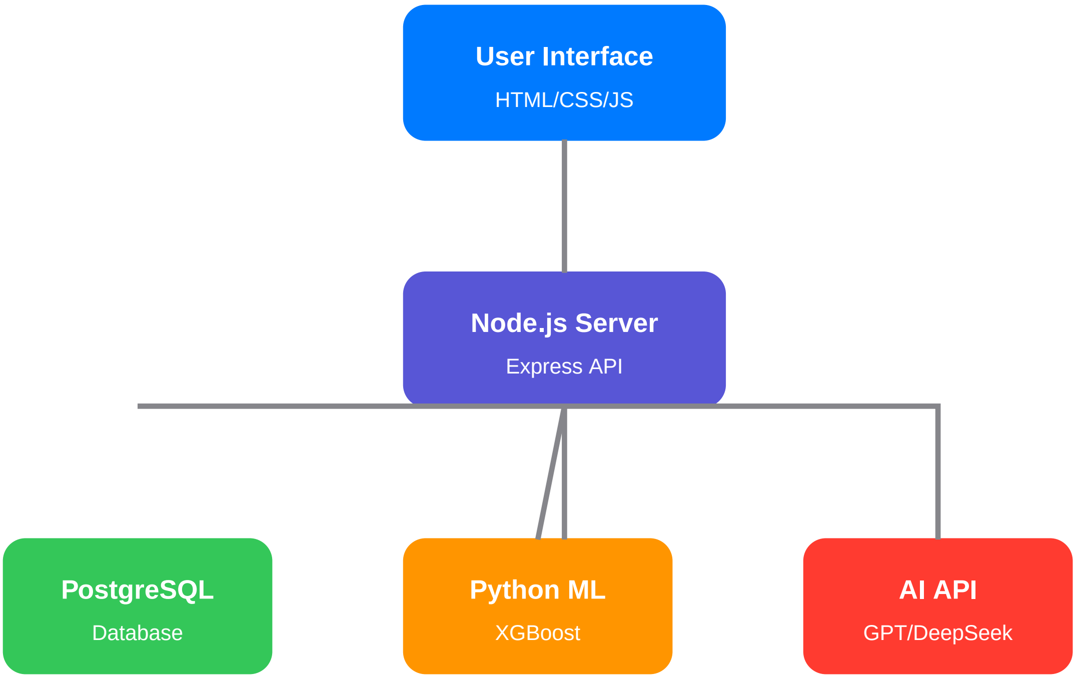
## 9.1 Target Performance Metrics

| Metric | Target | World Best | Status |
|---|---|---|---|
| **MAE** | < 2.5 patients | 2.63-2.64 | In Progress |
| **MAPE** | < 2.5% | ~2-3% | In Progress |
| **Directional Accuracy** | > 93% | ~91% | In Progress |
| **80% CI Coverage** | > 80% | ~85% | In Progress |
| **95% CI Coverage** | > 95% | ~95% | In Progress |
| **R-squared** | > 0.97 | ~0.95 | Planned |

## 9.2 Evaluation Metric Formulas

| | |
|---|---|
| **MAE** | MAE = (1/n) * SUM(\|y_i - y_hat_i\|) |
| **MAPE** | MAPE = (100/n) * SUM(\|y_i - y_hat_i\| / y_i) |
| **RMSE** | RMSE = sqrt[(1/n) * SUM((y_i - y_hat_i)^2)] |
| **R-squared** | R^2 = 1 - SS_res / SS_tot |

# 10 System Architecture




## Data Flow

1. User accesses web page, triggers prediction request
2. Node.js server receives request
3. Fetch historical data from PostgreSQL
4. Call Python XGBoost model (if available)
5. Call AI API for real-time event analysis
6. Combine all factors to calculate final prediction
7. Return prediction result with confidence intervals

## Technology Stack

| | |
|---|---|
| **Frontend:** | HTML5, CSS3, JavaScript (ES6+), Chart.js |
| **Backend:** | Node.js 18+, Express |
| **Database:** | PostgreSQL 15+ |
| **ML:** | Python 3, XGBoost, NumPy, Pandas |
| **AI:** | OpenAI GPT, DeepSeek |

**Deployment:**

Railway, Docker

# Conclusion

The NDH AED Prediction System is a world-class prediction platform that combines statistics, machine learning, and artificial intelligence. By integrating multiple advanced techniques and methods, the system delivers highly accurate emergency department attendance predictions, helping hospital management with effective resource planning and staff allocation.

## Core Advantages

- Multi-factor multiplicative model - considers time, weather, holidays, AI factors
- Dynamic factor calculation - rolling window with exponential decay weights
- Machine learning enhancement - XGBoost captures complex nonlinear patterns
- Real-time AI analysis - automatically identifies and quantifies news events
- Multiple smoothing methods - 9 techniques for robust final predictions
- Uncertainty quantification - confidence intervals to support decision-making

## Future Development

1. Integrate more external data sources (flu surveillance, air quality)
2. Multi-horizon prediction (1-6 hours, 1-7 days, 1-4 weeks)
3. Develop admitted patient prediction functionality
4. Continuous algorithm optimization to achieve world-best accuracy
5. Publish academic papers for international recognition

We are committed to making the NDH AED Prediction System the most accurate and reliable emergency department attendance prediction tool in the world.

**North District Hospital**

Hospital Authority, Hong Kong

Version 2.5.4 | December 2025