Quality-aware wireless urban sensor networks using deep-learning

Environment and Planning B: Urban Analytics and City Science
XX(X):2–33
©The Author(s) 2024
Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

Carrow Morris-Wiltshire, Stuart Barr, Phil James and Tom Komar

Abstract

Data quality issues in Wireless Sensor Networks (WSNs) present significant challenges for smart city applications, particularly in pedestrian monitoring systems where complex computer vision algorithms must process video streams to derive count data. While extensive research exists on WSN data quality in environmental monitoring, there is limited understanding of its impact on predictive modelling in urban mobility applications such as pedestrian dynamics. This paper investigates the relationship between data quality and performance of models predicting the number of pedestrians in a given location using sensor data provided by the UK's Newcastle Urban Observatory. We develop a quality-aware machine learning pipeline that processes real-time pedestrian count data for emergency response and resource monitoring applications. Univariate and multivariate approaches using Long Short-Term Memory (LSTM) models are compared. Analysis reveals that the length of consecutive data sequences has a greater impact on model performance than overall data completeness, which provides valuable insight into data management strategies. Notably, sensors with average sequence lengths below 20 hours showed higher prediction errors despite better data completeness, attributed to their inability to capture daily periodicities in pedestrian movement patterns. It is shown that for multivariate models encoding periodicities such as the daily, weekly, and annual cycle, can improve prediction accuracy over univariate approaches, particularly for longer forecasting horizons. These findings advance our understanding of the minimum data quality requirements for reliably predicting the dynamics of the Urban Observatory pedestrian data, and provide valuable insights for developing the robust quality-aware systems for real-time emergency response and urban resource management.

Keywords

Wireless Sensor Networks, Pedestrian Monitoring, Data Quality, Machine Learning, Movement Prediction

Newcastle University, UK

Corresponding author:
Carrow Morris-Wiltshire, Newcastle University, UK
Email: c.morris-wiltshire@ncl.ac.uk

Prepared using sagej.cls *[Version: 2017/01/17 v1.20]*

Introduction

The Internet of Things (IoT) and Wireless Sensor Networks (WSNs) have become fundamental components of smart city infrastructure, enabling real-time monitoring and analysis of urban environments (Akyildiz et al. 2002; Atzori et al. 2010; Smith and Turner 2019). WSNs consist of spatially distributed autonomous sensors that cooperatively monitor physical or environmental conditions, transmitting data wirelessly to central gateways for processing and analysis (Gubbi et al. 2013). Data quality in WSNs is critically important for making informed decisions about urban dynamics (Klein and Lehner 2009). Current cities typically have limited decision-making functionality due to slow, incomplete, and largely disconnected information transmission systems (Barr et al. 2020).However, as cities deploy more extensive networks of low-cost sensors to improve spatial resolution, they face increased data quality challenges (Karkouch et al. 2016). Low-cost sensors are more prone to errors and inconsistencies than their expensive counterparts, and the volume of data generated makes manual quality assurance infeasible (Teh et al. 2020). This necessitates automated systems that are 'quality-aware' and capable of assessing and managing data quality in real-time (Sarrab et al. 2020; Buelvas et al. 2023). Real-time streaming scenarios present particular challenges, as data inconsistencies can lead to suboptimal or incorrect decisions in automated systems (Klein and Lehner 2009). For example, a sensor counting pedestrians might fail to detect an overcrowding event due to a malfunction, potentially leading to a delayed emergency response. To address these challenges, systems must integrate quality control mechanisms throughout their data pipelines while providing users with transparent insights into data provenance (Elkhodr and Alsinglawi 2020).

While significant research exists on data quality for WSNs in environmental monitoring applications (Van Zoest et al. 2021; Buelvas et al. 2023; Şimsek et al. 2024), there remains a distinct gap in understanding data quality requirements for complex dynamic systems such as urban mobility (Gorshenin et al. 2024). Urban mobility represents a complex adaptive system of systems, characterised by intricate interactions between pedestrian flows, transportation networks, and urban infrastructure (Batty 2007). The intellectual challenge lies in accurately predicting the spatio-temporal dynamics of these interconnected systems in real-time - a task made particularly difficult by the non-linear nature of human

movement patterns and the inherent variability in sensor data quality (Klein and Lehner 2009; Karkouch et al. 2016). This predictive capability is a key enabling technology for urban digital twins, which promise to revolutionise city management through real-time emergency response, infrastructure maintenance optimisation, and resource monitoring (Mohammadi and Taylor 2017; Dembski et al. 2020).

The critical prerequisite for reliable prediction is understanding the degree to which the sensor network data can be trusted, requiring real-time 'quality awareness' of incoming data streams (Elkhodr and Alsinglawi 2020; Teh et al. 2020). This is particularly challenging in urban environments where sensor networks generate high-velocity, high-volume data streams subject to various quality issues including missing values, noise, and systematic biases (Karkouch et al. 2016). Without automated mechanisms to assess and validate data quality in real-time, the development of trustworthy predictive models - and by extension, effective real-time decision-making systems - remains fundamentally constrained Elkhodr and Alsinglawi (2020). This paper begins to address this gap by investigating the relationship between data quality dimensions (different measures of data quality such as completeness) and the performance of a model predicting complex spatio-temporal dynamics of pedestrian movements (Teh et al. 2020).

Literature Review

Current approaches to WSN data quality (DQ) management reveal significant limitations when applied to urban mobility prediction. Traditional frameworks, such as those proposed by Wang and Strong (1996) and later refined by Karkouch et al. (2016), emphasise static quality dimensions like accuracy and completeness but fail to adequately address the temporal dependencies crucial for movement prediction. While these frameworks provide valuable taxonomies for categorising DQ issues, they offer limited practical guidance for handling the dynamic nature of urban mobility data. For instance, Che et al. (2018) demonstrate that conventional data cleaning approaches, which prioritise completeness through simple imputation methods, often fail to preserve the temporal patterns essential for accurate prediction. The approach suggested by Klein and Lehner (2009) involves using DQ control mechanisms that adjusts the size of the data window based on the 'interestingness' of the data, allowing for adaptability to the data's

temporal dynamics. However even with these dynamic adjustments, the limited temporal context available in an adjustable window makes it unsuitable for capturing the complex, non-linear relationships inherent in urban mobility data.

The limitations of existing approaches become particularly evident in real-time applications. Current quality control mechanisms, as reviewed by Teh et al. (2020), typically rely on post hoc detection and correction of DQ issues, making them unsuitable for real-time decision support systems. Furthermore, most existing studies have focused on synthetic datasets or controlled environments (Shukla and Sengupta 2020; Nguyen et al. 2021), leaving a significant gap in understanding how these approaches perform with the messy, incomplete data streams characteristic of real-world urban sensor networks. This gap is particularly problematic for pedestrian monitoring systems, where DQ issues can arise from various sources including environmental conditions, sensor malfunctions, and communication failures (Elkhodr and Alsinglawi 2020).

Accurate predictive modelling plays a critical role in establishing quality-awareness around data from WSNs because it enables a deeper understanding of the expected behaviour of the data Nguyen et al. (2021); Aouedi et al. (2025). This understanding is essential for identifying deviations from normalcy, which may indicate data quality issues (Martín-Chinea et al. 2023). By comparing the actual sensor readings to the predictions generated by the model, one can identify anomalies or outliers that deviate significantly from the expected pattern - these deviations can signal potential data quality problems, such as sensor malfunctions, environmental interference, or data transmission errors Nguyen et al. (2021).

Teh et al. (2020) finds that the most common approaches to error detection and fault correction use artificial neural networks (ANNs). Aouedi et al. (2025) find that traditional methods such autoregressive integrated moving average (ARIMA) struggle to model the non-linearity and complex dependencies often found in traffic data, leading to lower prediction accuracy, especially for longer-term forecasting. Liu et al. (2023) compare performance of deep learning models for traffic forecasting, including LSTMs and transformers. They find that while transformers achieved higher prediction accuracy, they generally have a larger number of parameters (millions) compared to LSTMs (hundreds of thousands),

stating that transformer-based models can "suffer from long training and inference times".

LSTMs excel at capturing long-range dependencies due to their inherent memory mechanism, allowing them to process entire sequences of data and retain information from earlier time steps Nguyen et al. (2021). Sherstinsky (2020) find LSTMs can handle irregular sampling patterns and missing data points, which are common challenges in WSNs (Shukla and Sengupta 2020). This robustness to data irregularities makes LSTMs more reliable in real-world deployments where data quality can be compromised by sensor failures or network disruptions (Chen et al. 2021; Aouedi et al. 2025). Additionally, unlike traditional approaches LSTMs can be adapted to handle multivariate time series data, allowing them to learn from multiple sensor readings and capture the interdependencies between them (Nguyen et al. 2021). This capability is crucial for gaining a holistic understanding of the system being monitored and for improving the accuracy of anomaly detection (Nguyen et al. 2021; Aouedi et al. 2025). Greff et al. (2017) highlights that while several LSTM variants exist, none consistently outperform the standard LSTM architecture. Optimising LSTM hyperparameters, such as the number of layers, cells, and learning rate, can significantly impact computational efficiency and techniques like model pruning or quantisation can further reduce computational demands without significantly sacrificing accuracy (Aouedi et al. 2025).

Methodology

Data Collection

The Newcastle Urban Observatory, one of the largest public-facing urban sensing networks in the United Kingdom, provides real-time monitoring of various urban metrics across Newcastle upon Tyne. The network of pedestrian monitoring devices deployed across Newcastle is illustrated in Figure 1. Each monitoring station integrates three key components: an IP-camera for video capture, a Raspberry Pi processing unit enhanced with a Google Coral TPU AI-accelerator, and a cellular modem for data transmission (Komar and James forthcoming). The system conducts real-time video analytics, performing object detection and tracking while recording line-crossing events. This processed data is then

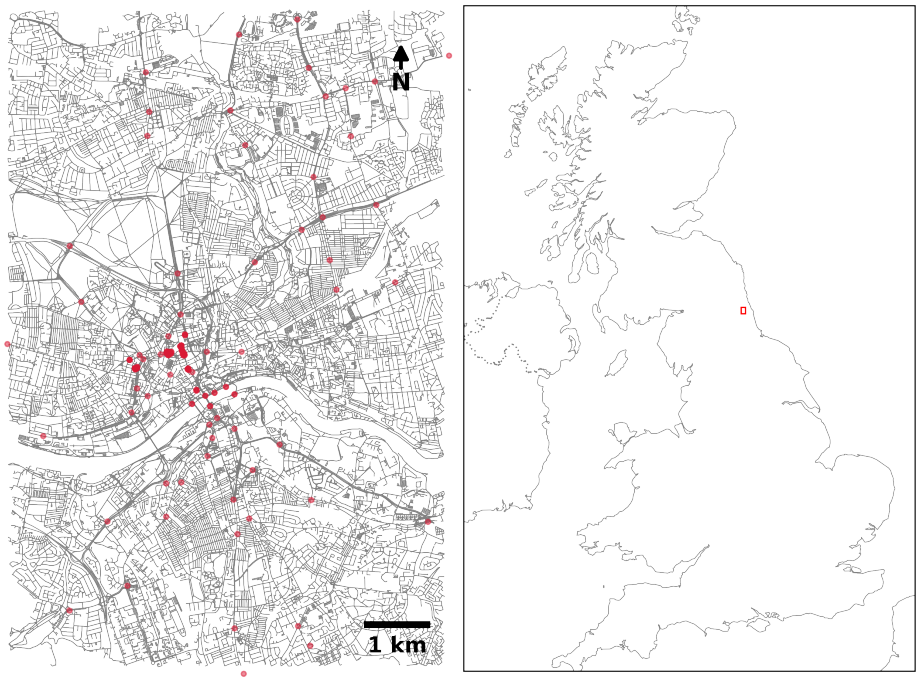

Figure 1. Distribution of sensors at Newcastle upon Tyne's Urban Observatory

transmitted to a central backend system, which serves as our primary data source for subsequent analysis.

The data from the pedestrian sensors is aggregated at 15-minute intervals, with each record containing the number of pedestrians detected within a scene over that time period. Figures 2 shows the raw data from a single sensor over 18 days, highlighting two distinct patterns of missing data: regular nighttime gaps (a design feature of some sensors) and extended periods of data sparsity due to technical issues such as sensor failure or data transmission problems. Figure 3 quantifies this data incompleteness, showing that while there should ideally be 96 records per day (4 per hour), few days achieve complete coverage and some days have no records at all.

To more clearly indicate the extent of the missing data points, Figure 3 shows the percentage of records recorded per day for the same sensor. The plot shows that

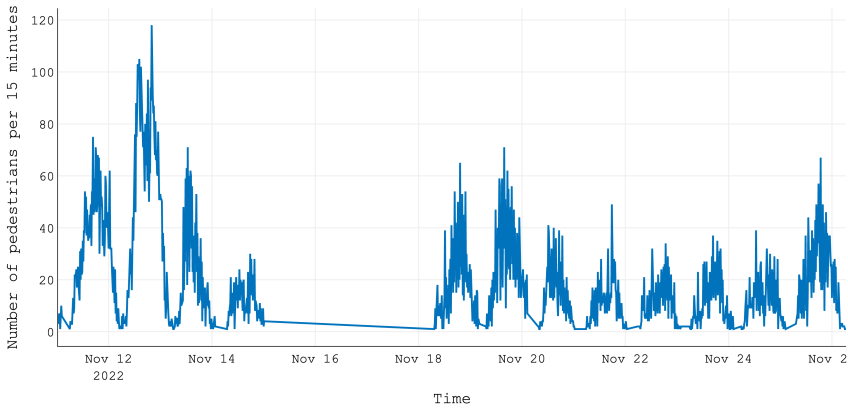

Figure 2. Example of preprocessed data from a sensor. Short gaps can be seen where zero values would be expected, and a longer gap can be seen where the data is missing.

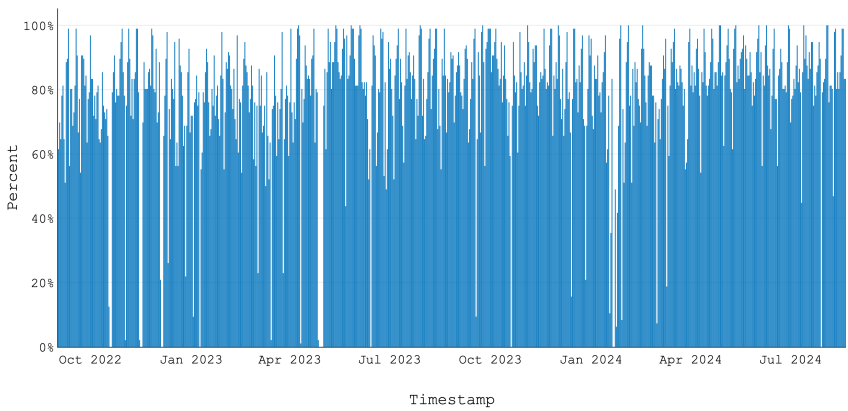

Figure 3. Example of the percentage of records recorded per day a for specific sensor.

there are many missing records over the 2-year period, and very few days with 100% data coverage. It is also evident from Figure 3 that there are a few days where no records are present at all.

Data Preprocessing

The data preprocessing stage executes functions related to data cleaning (data removal, data interpolation, and data sequencing), that are necessary before any feature engineering takes place. Whilst there are deep learning models that can handle missing data, these models tend to be more complex and tuning them can be challenging (Che et al. 2018). Another option is using complex imputation methods to fill in the missing data, but these are often costly and can affect the integrity of the data (Che et al. 2018). For these reasons, the data has been left in its raw form, and a consecutive sequence detection and labelling algorithm has been developed. Consecutive sequences are sections of the time series that exist without any missing data. The algorithm is shown in pseudocode (Algorithm 1). It works by scanning through a DataFrame row by row, building sequences where the time difference between consecutive rows equals the recording interval of the sensor (normally 15-minutes). When there's a gap in the timestamps or when, it checks if the current sequence meets the minimum length requirement (window_size + horizon - see Figure 6 for a visualisation of the parameters). Valid sequences are stored and numbered sequentially.

Two key hyper-parameters define the window structure (Längkvist et al. 2014):

1. *window_size*: Determines how many historical time steps are included in each window. This parameter can be tuned to optimise model performance.

2. *horizon*: Specifies how far into the future the model predicts. This parameter is not tuned and is set based on the desired prediction length.

Figure 4 shows the outputs of the data preprocessing stage for the same sensor as Figures 2 and 3 where n is the temporal index of each observation and *value* is the number of pedestrians observed. Each new consecutive sequence is shown as a new colour. The sequences in this example capture similar segments of the daily cycle before a missing record appears. This means each sequence is less than 96 records long which limits the length of training windows that can be used in the machine learning pipeline (for example a window size of 24 with a horizon of 24 requires at least 48 consecutive time steps, otherwise the sequence will be discarded).

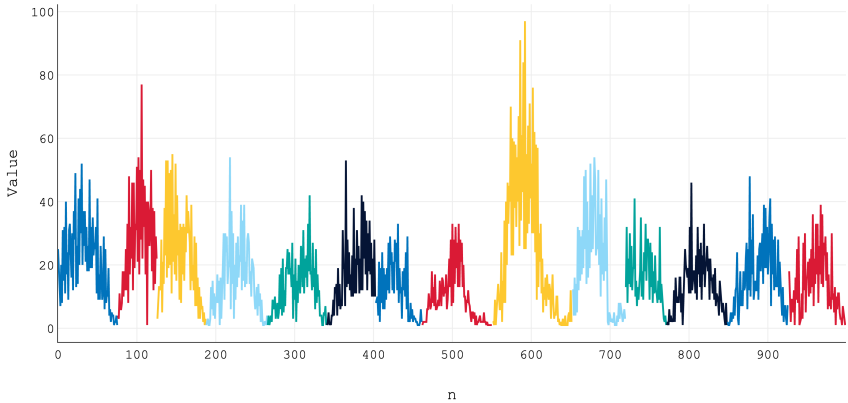

Figure 4. Example of preprocessed data from a sensor. Each colour shows a new sequence.

Algorithm 1 Find and process consecutive sequences

```
Require: DataFrame df with time series data
Ensure: DataFrame with consecutive sequences and assigned sequence numbers
function ProcessConsecutiveSequences(df)
    sequences ← ∅
    current_sequence ← ∅
    Δt_min ← minimum time delta between timestamps
    w ← window size
    h ← horizon
    for each row r in df do
        if current_sequence = ∅ or time difference between r and last row of
    current_sequence = Δt_min then
            Add r to current_sequence
        else
            if length of current_sequence > w + h then
                Add current_sequence to sequences
            end if
            current_sequence ← {r}
        end if
    end for
    if length of current_sequence > w + h then
        Add current_sequence to sequences
    end if
    result ← ∅
    for i ← 1 to |sequences| do
        Assign sequence number i to all rows in sequences[i]
        Add sequences[i] to result
    end for
    return result
end function
```

Feature Engineering

The feature engineering stage tests the hypothesis that incorporating explicit temporal features alongside raw sensor data could improve prediction accuracy, particularly for longer horizons. This hypothesis proposes that combining harmonic decomposition through Lomb-Scargle periodograms (Press and Rybicki 1989; VanderPlas 2018) with deep learning would enable better capture of both cyclical patterns and irregular fluctuations in pedestrian movement. The proposed dual representation strategy - using both raw time series and engineered temporal features - was designed to provide a more robust foundation for handling the inherent periodicities in urban mobility data while maintaining resilience to gaps in sensor observations. To test this hypothesis, the methodology specifically differentiates between short-term (2-hour) and longer-term (6-hour) prediction horizons to evaluate how feature complexity requirements scale with prediction length.

The features are calculated using a Lomb-Scargle Periodogram (LSP) (VanderPlas 2018). The LSP was selected over traditional Fourier Transform methods for several key reasons. While Fourier Transform is widely used for spectral analysis, it requires evenly sampled data to function correctly. In our pedestrian counting system, data gaps create inherently uneven sampling - both from scheduled downtime (e.g., nighttime sensor deactivation) and unplanned interruptions (e.g., sensor malfunctions or communication failures). The LSP is specifically designed to handle such irregularly sampled time series data. The LSP produces a power spectrum that identifies and quantifies the strength of periodic signals in the pedestrian count data. For each potential frequency (or period), the LSP computes a normalised power value that indicates how strongly that frequency is represented in the data. High power values suggest strong periodic patterns at that frequency, while low values indicate weak or absent periodicities. The frequencies with the highest power values are then selected as the periodic features.

The features are then normalised using the Standard Scaler equation $X_i' = \frac{X_i - \mu}{\sigma}$, where X_i is the original value, μ is the mean of the feature, and σ is the standard deviation of the feature. The standard scalar normalisation technique is chosen to ensure that any future data points falling outside of the existing range can be scaled appropriately as a min max scaler would not be appropriate in this case

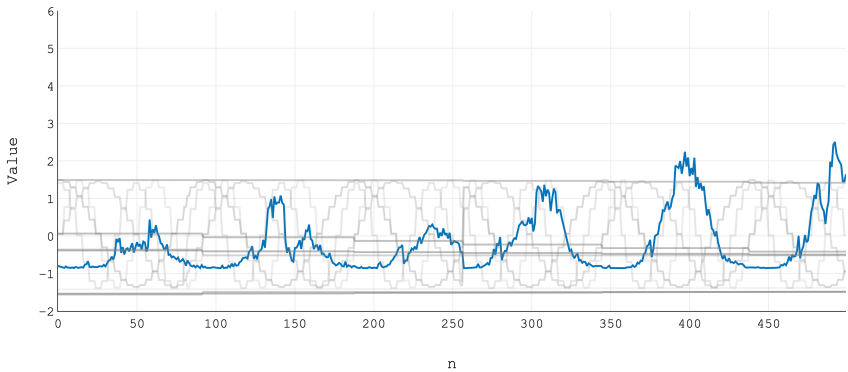

Figure 5. Example of feature engineering from a sensor for 500 data points where the 24hr and 12hr frequencies are most obvious. Frequency features are shown as grey lines and the pedestrian flow data as blue. The non-uniformity of the sinusoids highlights the missing data gaps.

(Hastie et al. 2001). Calculating the mean and standard deviation in the scaling process also serves as a check for future data drift; if movement patterns change over the time the mean is likely to be non-stationary. Figure 5 show an example of engineered frequency features from the same sensor as Figure 4 where engineered frequency features are shown as grey lines and the pedestrian flow data as blue (n is the temporal index of each observation and *value* is the scaled pedestrian count).

Data Loading

For recurrent neural networks (such as LSTMs), a number of approaches to data loading can be used. A common approach for time-series data is using fixed-sized windows (Pascanu et al. 2013; Yu et al. 2019). An example fixed-size window is shown in Figure 6 (n is is the temporal index of each observation and *value* is the scaled pedestrian count). A fixed-size window contains multiple sequences of data: the primary sensor readings (shown in blue) and additional engineering features (shown in grey). Each window is associated with a label (shown in red) which represents a future data point that the model aims to predict.

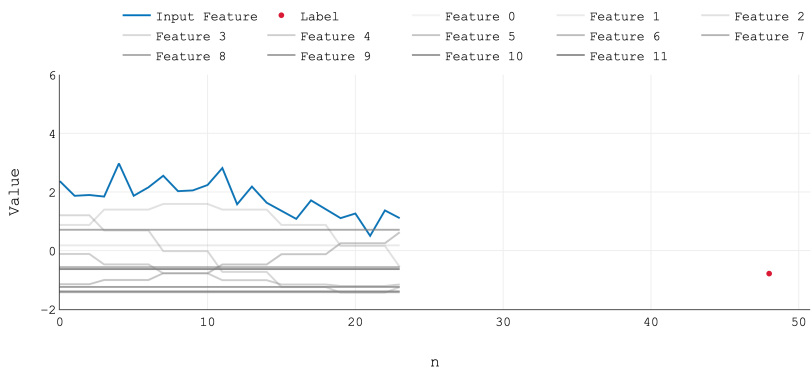

Figure 6. Example window used in the data loading process.

In the example plot, a window size of 24 and horizon of 24 are used. This means the model is trained to predict pedestrian flow 6 hours into the future using the previous 6 hours of data. The prediction is made recursively, using the standardised count values and the additional temporal features from within each window.

Model Training

Two primary experiments were conducted to evaluate model performance: an interpolation experiment and a multivariate/univariate (MV/UV) comparison. The interpolation experiment examined whether filling nighttime data gaps (between 00:00 and 06:00) with zero values would improve model performance. The multivariate/univariate comparison assessed whether incorporating additional engineered features, such as temporal patterns and frequency components (as illustrated in Figure 5), would enhance prediction accuracy compared to using only raw pedestrian counts. Both experiments measured comparative performance across twelve sensors selected for their superior data completeness (75%-85%).

For both experiments, we systematically tuned the models' hyper-parameters through an optimisation process consisting of 100 trials per experimental condition. The hyper-parameters and their possible values are presented in Table 1. These parameters include the window size (ranging from 4 to 48 time steps),

Table 1. Hyper-parameter tuning configuration

Parameter	Suggestion Type	Value Range/Options
window_size	suggest_categorical	[4, 8, 12, 16, 24, 32, 48]
batch_size	suggest_categorical	[32, 64, 128]
lr	suggest_loguniform	1e-5 to 1e-1
scheduler_step_size	suggest_int	3-7
model_type	suggest_categorical	["lstm", "gru"]
hidden_dim	suggest_categorical	[32, 64, 128, 256]
num_layers	suggest_int	1 to 3
dropout	suggest_uniform	0.0 to 0.5

batch size (32 to 128), learning rate (10^{-5} to 10^{-1}), scheduler step size (3 to 7 steps), model architecture (LSTM or GRU), hidden dimensions (32 to 256 units), number of layers (1 to 3), and dropout rate (0 to 0.5). While each experimental condition resulted in different optimal hyper-parameter values, we maintained consistent parameter ranges across all conditions to ensure a fair comparison.

To tune the models hyper-parameters, the Optuna library (Akiba et al. 2019) is used in conjunction with MLflow (Zaharia et al.) for experiment tracking. The hyper-parameter tuning process is shown in Figure 7 where S denotes a sensor and M denotes the associated model. The process involves running the pipeline multiple times with different hyper-parameters and recording the results. RMSE (Root Mean Square Error) was selected as the optimisation metric for hyper-parameter tuning, following established practices in time-series prediction tasks (Hyndman and Koehler 2006).

Model Evaluation

To evaluate the model a range of metrics are used, these are shown in Table **??**. The primary metric will be the root mean squared error (RMSE). RMSE measures the difference between the predicted and actual values and is more sensitive to outliers than the mean absolute error (MAE). Mean absolute percentage error (MAPE) will be used as it can show the relative error between predicted and actual values. R-squared (R^2) metric is also used. R^2 measures the proportion of the variance in the dependent variable that is predictable from the independent variable(s). The R^2 scores are expected to be low due to amount of noise in the data.

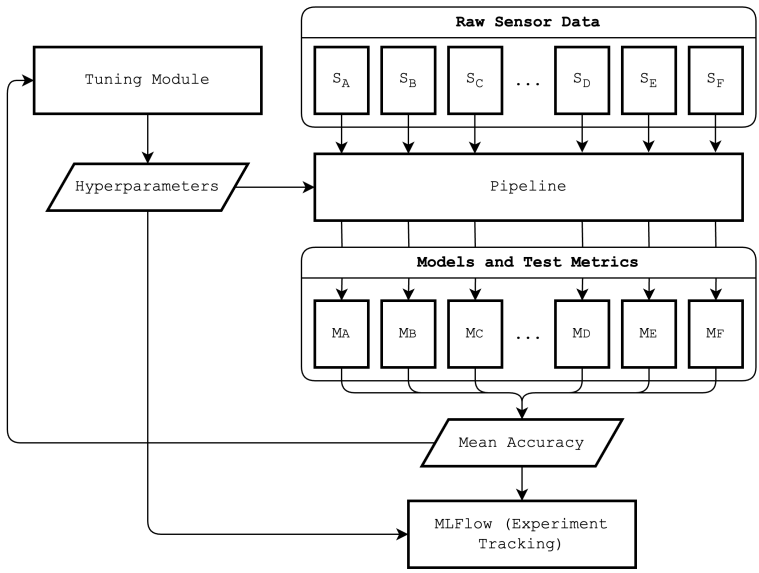

Figure 7. Hyper-parameter tuning process

Sensor data quality is evaluated using two primary metrics: average sequence length and data completeness for both experiments. Average sequence length, defined as the length of consecutive record sequences (detailed in Algorithm 1 and visualised in Figure 4), provides insight into data continuity. Data completeness is calculated as the ratio of actual records to expected records throughout the study period.

Hyper-parameter Tuning

The hyperparameter optimisation, detailed in Table 2, reveal distinct architectural differences among models. The UV-8 model emerged as the most compact, utilising a single layer with 64 hidden dimensions. In contrast, MV models maintained consistent architectures with 64 hidden dimensions across two layers. The UV-24 model proved to be the largest, implementing 256 hidden dimensions across two layers. This configuration may raise concerns about potential overparameterisation, as the UV-24 model contains four times the hidden units of the MV-24 model despite processing fewer input features (1 versus 12). Additionally, the MV-8 model's elevated dropout rate of 17.3% (compared to

5-7% in other models) suggests a higher susceptibility to overfitting, necessitating more aggressive regularisation.

Table 2. Comparison of tuned hyper-parameters for different experiments. The window size was consistently selected as 24 time steps (6 hours). All models selected the same batch size. Hidden dimensions and number of layers vary between experiments. The smallest model is UV8 and the largest is UV24.

	UV-24	MV-24	UV-8	MV-8
window_size	24	24	24	24
horizon	24	24	8	8
stride	1	1	1	1
batch_size	32	32	32	32
model_type	lstm	lstm	lstm	lstm
hidden_dim	256	64	64	64
num_layers	2	2	1	2
dropout	0.064	0.053	0.072	0.173
learning_rate	0.00011	0.00024	0.00382	0.00012
scheduler_step_size	7	4	4	3
scheduler_gamma	0.518	0.564	0.440	0.802

Results

Univariate / Multivariate Experiment

Tables 3 and 4 show the results for the 12 most complete sensors for the multivariate/univariate experiment. Where MV performs better than UV, the value is highlighted in yellow, and where UV performs better than MV the value is highlighted in teal. Analysis of these results show that for the smaller prediction horizon (2 hours), the univariate models generally outperformed the multivariate models across the 3 metrics of MAPE, RMSE and MAE. For the longer prediction horizon (6-hours) the multivariate models achieved marginally better performance metrics, exhibiting lower MAE and RMSE than their univariate counterparts (although the differences are very small). The difference in MAPE is still in favour of the univariate models. This pattern could suggest either an insufficient capture of temporal dependencies in the univariate models or a reliance on periodicity features for prediction. The strong performance of the univariate approach for the 2-hour models may be attributed to the effectiveness of recent historical values as predictors, whereas introducing multivariate features may decrease the signal to noise ratio.

Table 3. Comparison of test metrics for MV and UV models with a horizon of 8 timesteps (2 hours).

Test MAPE			Test RMSE			Test MAE		
UV	MV	diff	UV	MV	diff	UV	MV	diff
2.148	3.308	1.160	0.623	0.656	0.033	0.484	0.515	0.031
1.632	1.922	0.290	0.752	0.683	−0.069	0.604	0.542	−0.063
1.755	2.047	0.292	0.701	0.680	−0.021	0.528	0.515	−0.013
1.650	2.317	0.667	0.632	0.683	0.051	0.485	0.549	0.064
1.332	1.509	0.177	0.639	0.639	0.000	0.478	0.474	−0.003
16.165	13.558	−2.607	0.384	0.387	0.003	0.282	0.283	0.002
1.123	1.174	0.051	0.380	0.373	−0.007	0.270	0.275	0.005
2.221	2.504	0.283	0.405	0.464	0.059	0.304	0.366	0.062
0.865	0.755	−0.110	0.414	0.397	−0.017	0.297	0.310	0.012
1.466	1.869	0.403	0.603	0.614	0.011	0.458	0.479	0.021
1.446	2.289	0.843	0.528	0.605	0.077	0.407	0.476	0.068
1.986	2.212	0.226	0.748	0.681	−0.067	0.600	0.531	−0.069

Table 4. Comparison of test metrics for MV and UV models with a horizon of 24 timesteps (6-hours).

Test MAPE			Test RMSE			Test MAE		
UV	MV	diff	UV	MV	diff	UV	MV	diff
1.623	1.945	0.322	0.696	0.671	−0.025	0.546	0.535	−0.011
2.790	3.648	0.858	0.928	0.830	−0.098	0.760	0.680	−0.081
2.463	2.687	0.224	0.680	0.675	−0.005	0.509	0.520	0.011
1.431	1.442	0.011	0.665	0.610	−0.055	0.506	0.465	−0.040
15.748	15.547	−0.201	0.629	0.602	−0.027	0.470	0.448	−0.021
1.653	1.270	−0.383	0.492	0.442	−0.050	0.369	0.371	0.001
1.324	1.180	−0.144	0.450	0.356	−0.094	0.315	0.271	−0.044
1.321	0.980	−0.341	0.533	0.431	−0.102	0.400	0.359	−0.041
0.827	0.930	0.103	0.437	0.356	−0.081	0.313	0.277	−0.035
17.913	22.985	5.072	0.593	0.594	0.001	0.465	0.479	0.015
1.646	2.990	1.344	0.513	0.667	0.154	0.390	0.547	0.156
3.008	4.392	1.384	0.861	0.848	−0.013	0.681	0.668	−0.013

Examining the relationship between the data quality metrics of completeness and sequence length help to establish a relation between model performance and the quality of the training data from the sensor. Figures 8 and 9 present the relationship between Root Mean Square Error (RMSE) and these metrics, respectively. Each bar corresponds to a sensor, ordered by decreasing RMSE. The completeness values for these sensors vary between 0.75 and 0.85 and the RMSE values are between 0.35 and 1.0. From the graph it appears that sensors with less

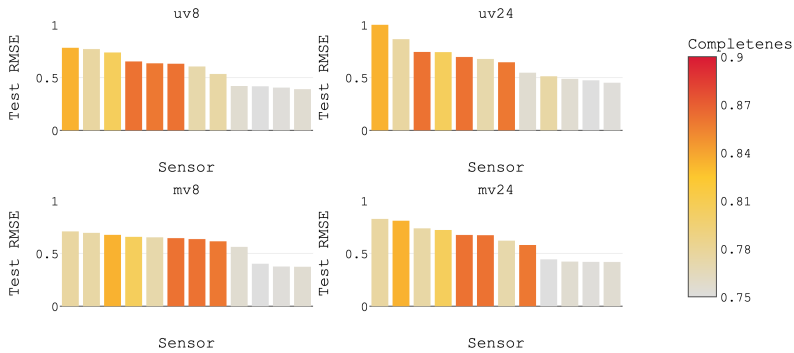

Figure 8. RSME vs Data Completeness for the 12 sensors with the most complete data.

data (lower completeness) have lower RMSE. Whilst this appears counterintuitive - one would expect that more data would lead to better model performance Figure 9 provides important context that explains this pattern. The average sequence lengths vary from 80 to 150 and the RMSE between 0.35 and 1.0. The sensors with the highest RMSE have the shortest average sequence length of around 20 hours which are the sensors with the highest data completeness. The lowest RMSE values are observed for sensors with an average sequence of around 35 hours. The distinct difference in RMSE values between sensors with different average sequence lengths could be due to the model's inability to capture the main temporal dependencies in the data. Observing the Lomb-Scargle periodograms of the sensors (Figure 10), it is clear that the highest periodicity occurs at around 24 hours which corresponds to daily patterns of in pedestrians activity (morning and evening commutes, lunchtime etc.). It seems plausible that the sensors with <20 hours average sequence length are not capturing the main periodicity in the data, leading to higher RMSE values, despite having more records to train on.

‘’

Interpolation Experiment

For the interpolation experiment, missing values between 00:00 and 06:00 were replaced with zeros. These specific hours were chosen based on historical

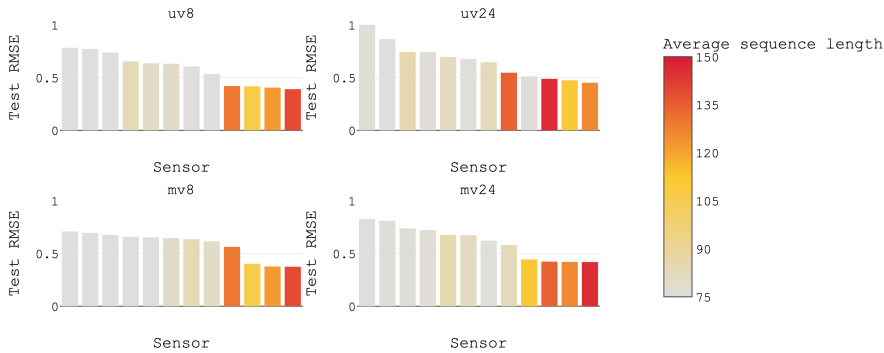

Figure 9. RSME vs Average Sequence Length for the 12 sensors with the most complete data.

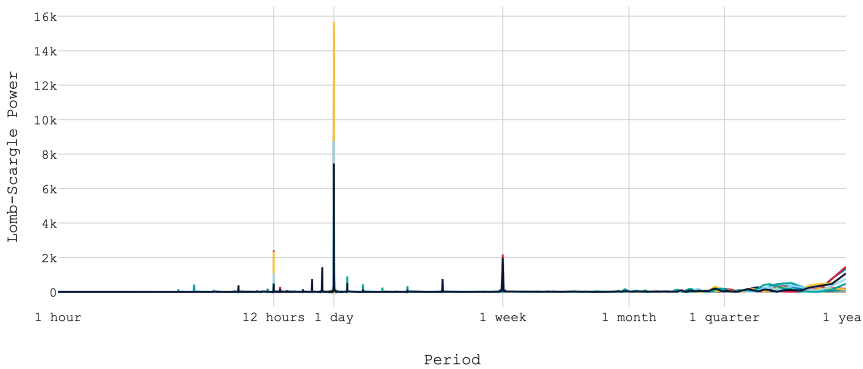

Figure 10. Lomb-Scargle periodograms showing the dominant periodicities in pedestrian flow across all sensors. The x-axis shows the period length in hours, and the y-axis shows the normalised spectral power, indicating the relative strength of each periodic component.

pedestrian activity in the city centre, which typically show minimal footfall traffic during these early morning hours, making zero values a reasonable approximation for most locations. As many of the sensors were programmed to not record when zero pedestrians were detected, the idea was to see if the short sequence lengths

associated with these sensors could be remedied. Figure 11 illustrates the resultant effect on average sequence length, demonstrating a substantial increase from 20–40 hours to 80–200 hours. However, as evidenced in Table 5, this interpolation yields negligible improvements in model RMSE. The limited effectiveness can be attributed to the simplistic nature of the interpolation method, which assumes zero pedestrian activity during nighttime hours across all sensors. This assumption proves particularly problematic for sensors that experience significant nighttime activity *and* extensive missing periods, as it introduces artificial patterns of zero values into the training data during extended missing periods. A potential refinement would involve expanding the training window from 24 time steps (6 hours) to 48 time steps (12 hours), thereby enabling the model to better contextualise isolated patterns of zero values.

Table 5. Comparison between raw and interpolated data for **multivariate** models with a horizon of 24 hours.

Sensor Name	Test MAE			Test Std.		
	Raw	Int	diff	Raw	Int	diff
GREYSTTHEATRESOUTH_NORTHWEST_TO_SOUTHWEST	0.547	0.584	0.037	0.533	0.656	0.122
GREYSTTHEATRESOUTH_SOUTHWEST_TO_NORTHWEST	0.479	0.597	0.118	0.609	0.760	0.150
NCLNORTHUMBERLANDSTSAVILLEROWEAST_NORTHEAST_TO_SOUTHEAST	0.277	0.336	0.058	0.390	0.513	0.123
NCLNORTHUMBERLANDSTSAVILLEROWEAST_SOUTHEAST_TO_NORTHEAST	0.359	0.359	0.000	0.514	0.555	0.041
NCLNORTHUMBERLANDSTSAVILLEROWWEST_NORTHWEST_TO_SOUTHWEST	0.271	0.324	0.053	0.400	0.495	0.095
NCLNORTHUMBERLANDSTSAVILLEROWWEST_SOUTHWEST_TO_NORTHWEST	0.371	0.336	-0.034	0.514	0.513	-0.001
NCLPILGRIMSTMARKETLN_FROM_NORTH_TO_SOUTH	0.448	0.466	0.017	0.650	0.675	0.025
NCLPILGRIMSTMARKETLN_FROM_SOUTH_TO_NORTH	0.465	0.500	0.034	0.702	0.721	0.019
NCLSIDESTCROWNPOSADANORTH_FROM_EAST_TO_WEST	0.668	0.886	0.218	0.951	0.904	-0.047
NCLSIDESTCROWNPOSADANORTH_FROM_WEST_TO_EAST	0.520	0.612	0.092	0.745	0.822	0.076
NCLSIDESTCROWNPOSADASOUTH_FROM_EAST_TO_WEST	0.680	0.558	-0.122	0.905	0.752	-0.154
NCLSIDESTCROWNPOSADASOUTH_FROM_WEST_TO_EAST	0.535	0.496	-0.039	0.752	0.649	-0.103

Predictive Modelling Performance

Following these experimental outcomes, the feasibility of predicting model performance using the data quality metrics was investigated. The ability to predict model performance from data quality metrics would provide valuable guidance for sensor network deployments, helping organisations assess whether their data collection methods are likely to yield reliable predictive models before investing significant resources in model development. A Random Forest regressor with K-fold cross-validation to predict the Mean Absolute Error (MAE) of the LSTM

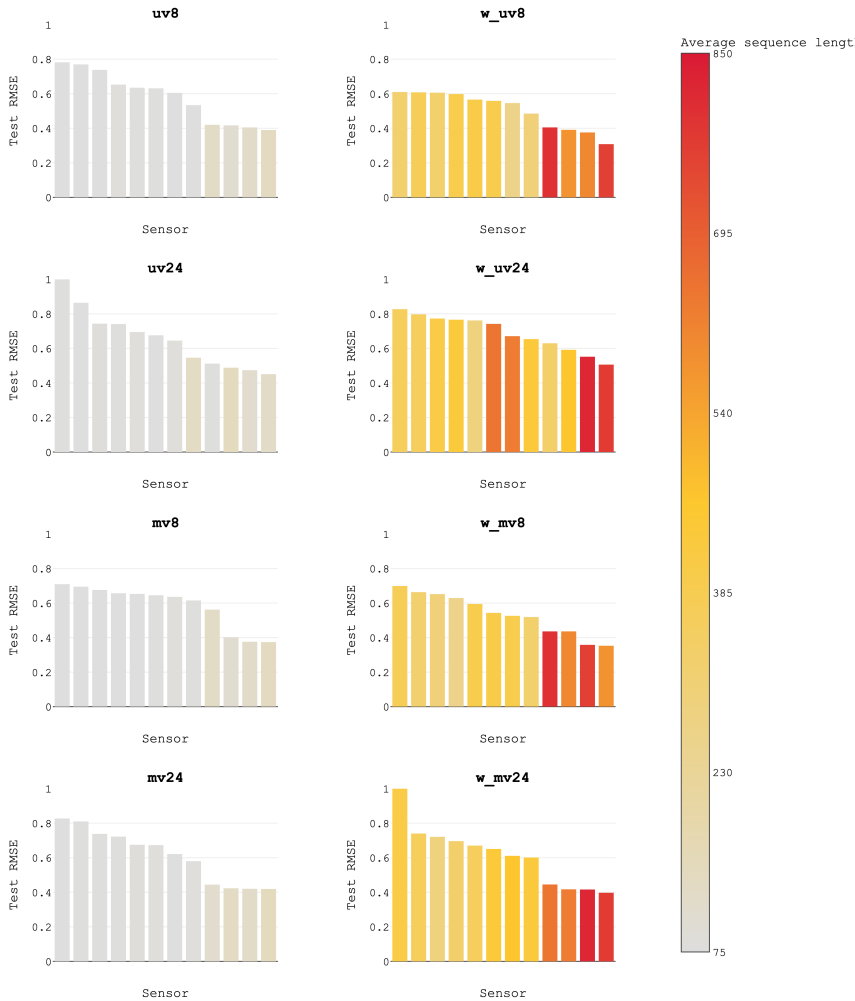

Figure 11. Comparison of average sequence length after the preprocessing stage for each of the 8 variations.

models was used with predictor variables, identified through feature importance analysis (Breiman 2001; Pedregosa et al.), comprising of average sequence length, data completeness, sequence count, and the standard deviation of sequence

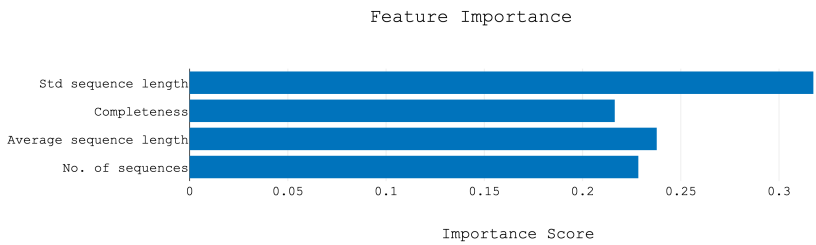

Figure 12. MV24 Random Forest feature importance.

lengths. Figures 12, 13, and 14 present the results of this analysis. While the model achieved an R-squared value of 0.3, indicating modest correlation, this performance metric should be interpreted cautiously due to potential overfitting on the limited sensor dataset. The four features all show relatively high importance in predicting the MAE of the LSTM models 12. 13 show the effect of each feature on the predicted MAE.

It may be that using R^2 for correlation is not the best metric for this analysis, as the relationship between the data quality metrics and the model performance is not necessarily linear. The partial dependence plots in Figure 13 show that the relationship between the data quality metrics and the model performance is not linear. For example, the average sequence length has a positive effect on the model performance up to a certain point, after which the effect is negative. This suggests that the relationship between the data quality metrics and the model performance is more complex than a simple linear relationship.

Discussion

Data Quality Assessment and Implications

The investigation into IoT pedestrian data streams reveals that data quality challenges stem from multiple factors, aligning with Karkouch et al. (2016)'s comprehensive analysis of IoT data quality issues. While their work established a broad framework for understanding IoT data quality, our findings specifically demonstrate the critical importance of temporal continuity in data sequences for predictive modelling performance. The relationship between sequence length

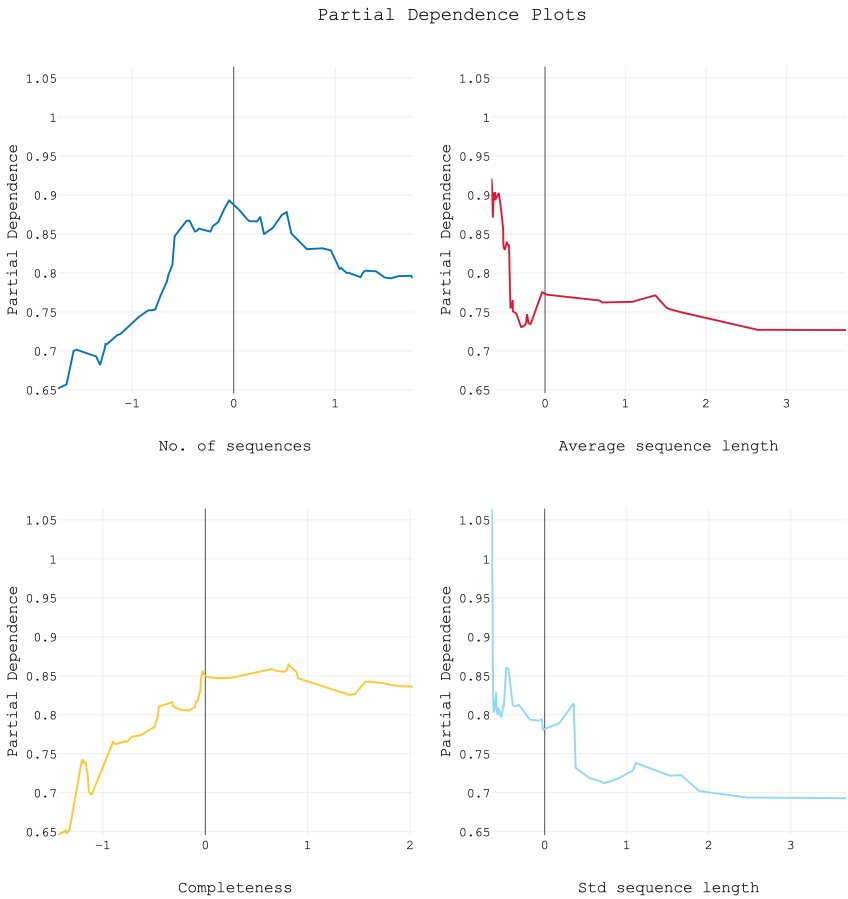

Figure 13. MV24 Random Forest partial dependence plots.

and model performance, particularly the threshold of 20 hours for capturing daily periodicities, addresses a gap in existing IoT data quality frameworks. This finding extends the traditional data quality taxonomies by emphasising the temporal dimension of data quality in urban sensing applications. Our analysis revealed that sensors with shorter average sequence lengths (<20 hours) exhibited poorer predictive performance despite higher overall data completeness rates. Sensors with >80% completeness achieved RMSE values >0.5 whereas the best performing sensors had 75% completeness. The same sensors with lowest completeness had the longest average sequence lengths (around 35 hrs for the 75% complete

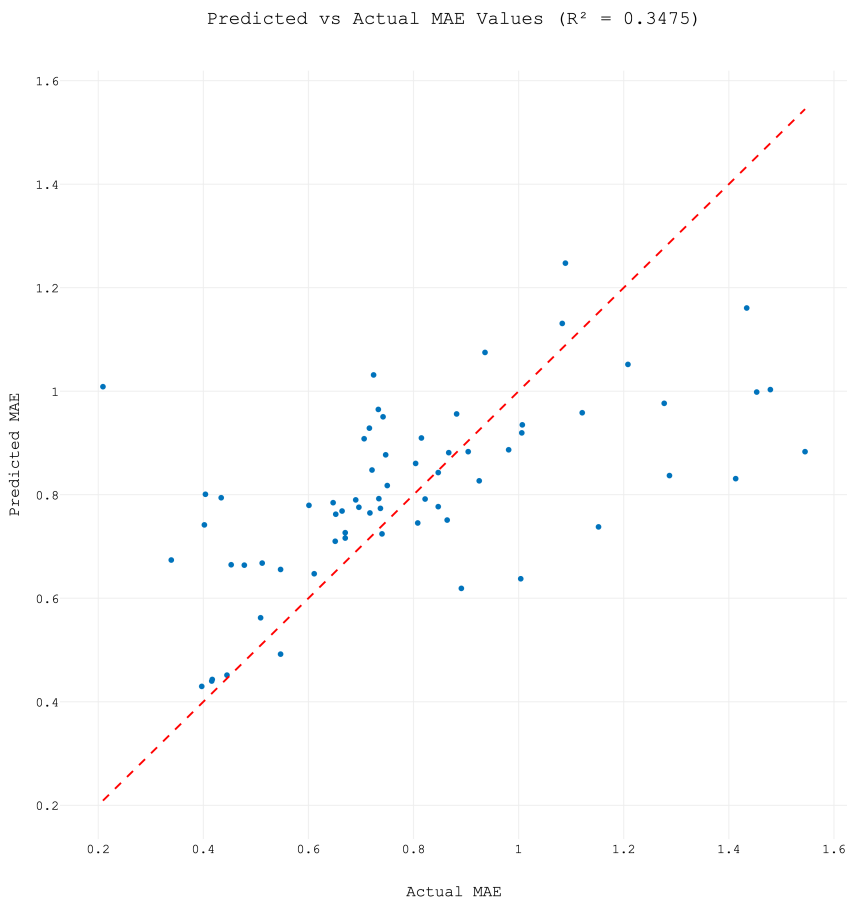

Figure 14. R-squared plot for predicted vs actual MAE values

sensors, compared with 20 hrs for the >80% complete sensors).This apparently counterintuitive result can be explained through the lens of Teh et al. (2020)'s work on temporal dependencies in sensor networks. The inability to capture complete diurnal cycles significantly impairs model performance, suggesting that traditional data quality metrics may need revision for time-series applications in urban environments.

The comparative performance of univariate and multivariate models provides insight into the relationship between data quality and model complexity. Our

findings align with Blázquez-García et al. (2021)'s observations regarding the importance of feature engineering in time-series analysis, while extending their work to specifically address urban pedestrian monitoring applications. The marginally better performance of multivariate models for longer prediction horizons, particularly in lower quality datasets, suggests additional features that capture some of the higher order temporal dependencies may help compensate for data quality issues. It is however unclear whether it is worthwhile including these multivariate features given the computational overhead of including engineered features in the training process. Further research will need to be done on larger prediction horizons to determine whether there a is a trend between performance, prediction horizon and the number of features included in the model. The average sequence lengths in the sensor data were insufficient to conduct these experiments using the existing fixed time window methodology. It seems plausible that for much longer prediction horizons (up to 24 hours) that the multivariate models would significantly outperform the univariate models making the feature engineering process worthwhile.

Attempts at simple interpolation proved largely ineffective, supporting Che et al. (2018)'s assertion that complex imputation methods might be necessary for meaningful improvement. Future work will need to investigate whether sophisticated interpolation can rival the performance of the sensors that capture full diurnal cycles.

There is some evidence to suggest that simple measures of data quality could be used to predict whether the data in a network of sensors is of sufficient quality to be used for predictive modelling for different prediction horizons. The framework that has been presented as part of this research could be used by organisations to ensure their sensor networks contain enough data to be used for predictive modelling, before expensive and time-consuming machine learning pipelines are developed.

Implementation, Limitations, and Practical Recommendations

The findings from this research have led to specific modifications in data collection protocols for the Urban Observatory's pedestrian sensors, particularly the implementation of completeness metadata and sensor heartbeat monitoring.

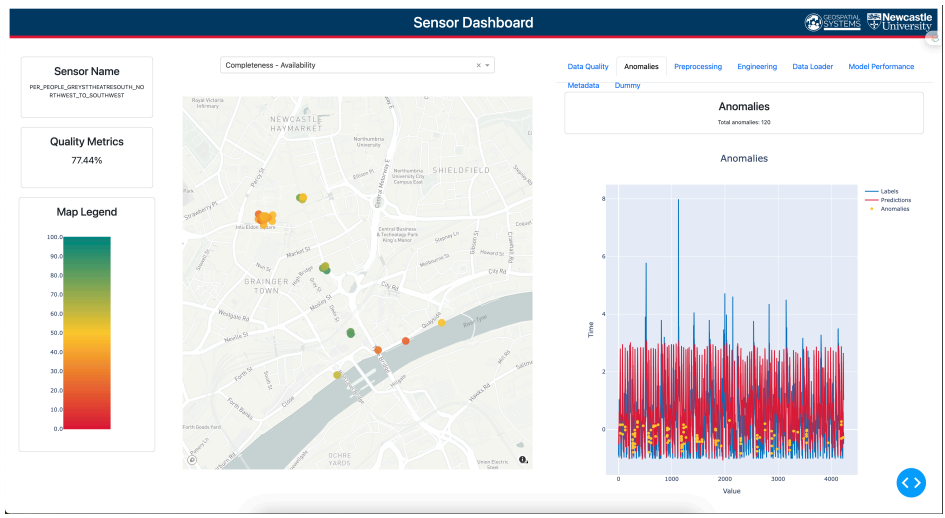

Figure 15. Screenshot of sensor dashboard showing completeness metric overlay.

Sensor heartbeat monitoring outputs a boolean value that allows us to differentiate between periods where the sensor is not outputting data because there is no activity and when the sensor is not working. These changes align with Elkhodr and Alsinglawi (2020)'s recommendations for enhanced data provenance in IoT systems, while addressing the specific challenges of pedestrian monitoring. The introduction of sensor heartbeat monitoring, in particular, represents a practical application of Cheng et al. (2018)'s proactive data quality management principles. Our quality-aware pipeline demonstrates the feasibility of automated quality assessment in real-time sensor networks, addressing a key challenge identified by Sarrab et al. (2020). A dashboard has also been developed to assess the performance of the library during development and experimentation. One of the features of the dashboard is to have a simple overview of data quality metrics like completeness and freshness shown in 15 and 16. The dashboard implementation, combining Flask and Dash frameworks, provides a scalable solution for real-time quality monitoring, though it currently lacks any of the sophisticated quality control mechanisms proposed by Klein and Lehner (2009).

This study faced several methodological constraints that should be considered when interpreting its findings. A primary limitation was the rapid drop-off in data availability for prediction horizons beyond 6 hours, stemming from the

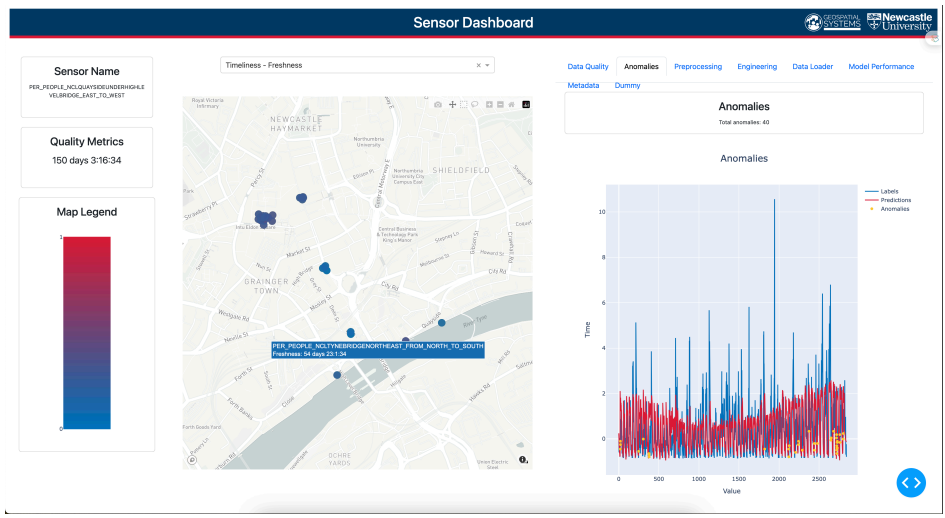

Figure 16. Screenshot of sensor dashboard showing freshness metric overlay.

fixed-window training approach and nighttime data gaps. The requirement for consecutive data windows meant that even a 6-hour prediction horizon needed at least 12-16 hours of continuous data when accounting for context and overlap. The approach of optimising for average performance across all sensors, while computationally efficient, potentially undermined opportunities for sensor-specific optimisation and may have resulted in overtraining for sensors with lower data completeness. Additionally, while our random forest model showed modest success ($R^2 = 0.3$) in predicting deep-learning model performance based on data quality metrics, these results should be interpreted cautiously given the limited sensor dataset. The absence of cross-fold validation means the model lacks exposure to the most recent 15% of data, potentially limiting its adaptability to evolving urban mobility patterns.

Summary

This study has revealed several insights about the relationship between data quality and predictive modelling performance in urban sensor networks. The analysis demonstrates that above a certain threshold of completeness (around 75% of the number of the total number of records) average sequence length plays an important role in model training accuracy - particularly when the average

sequence captures diurnal cycles. The investigation into multivariate versus univariate approaches suggests potential advantages of multivariate models for longer prediction horizons, though the benefits must be weighed against increased computational overhead. Furthermore, while simple interpolation techniques proved ineffective for improving model performance, the study established that basic data quality metrics could serve as indicators for determining whether sensor data is suitable for predictive modelling applications.

Future research should focus on addressing these limitations through several key avenues. The development of sensor-specific optimisation strategies would allow for more nuanced model tuning, potentially improving performance across diverse sensor configurations. Methods that utilise spatial dependencies between sensors could enhance prediction accuracy by leveraging the inherent relationships in pedestrian movement patterns across urban spaces. Integration of sophisticated anomaly detection algorithms, possibly incorporating multiple data quality dimensions, would enable more robust identification of sensor malfunctions and data quality issues, allowing these data points to be removed from a dataset, enhancing the quality of the training data. Finally, validation of these findings across larger sensor networks and diverse urban environments would strengthen the generalisability of the results and provide insights into scaling these approaches for broader smart city applications. These developments would further contribute to the robust and practical implementations of quality-aware predictive modelling systems that are critical in enabling real-time decision-making and management for urban environments.

Acknowledgements

The authors would like to thank the Newcastle Urban Observatory for providing access to their pedestrian monitoring data infrastructure. Special thanks to the Centre for Doctoral Training (CDT) in Geospatial Systems at Newcastle University for their academic support and guidance. We also acknowledge the valuable feedback received from colleagues in the School of Engineering and fellow CDT researchers.

Declaration of conflicting interests

The authors declare that there is no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Funding

This research was funded by the Defence Science and Technology Laboratory (DSTL) (RES/0539/7397) through the EPSRC Centre for Doctoral Training in Geospatial Systems (EP/S023577/1) at Newcastle University. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of DSTL or the UK Government.

References

Akiba T, Sano S, Yanase T, Ohta T and Koyama M (2019) Optuna: A Next-generation Hyperparameter Optimization Framework. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19. New York, NY, USA: Association for Computing Machinery. ISBN 978-1-4503-6201-6, pp. 2623–2631. DOI:10.1145/3292500.3330701.

Akyildiz IF, Su W, Sankarasubramaniam Y and Cayirci E (2002) Wireless sensor networks: A survey. *Computer networks* 38(4): 393–422.

Aouedi O, Le VA, Piamrat K and Ji Y (2025) Deep Learning on Network Traffic Prediction: Recent Advances, Analysis, and Future Directions. *ACM Comput. Surv.* DOI:10.1145/3703447.

Atzori L, Iera A and Morabito G (2010) The internet of things: A survey. *Computer networks* 54(15): 2787–2805.

Barr SL, Johnson S, Ming X, Peppa M, Dong N, Wen Z, Robson C, Smith L, James P, Wilkinson D, Heaps S, Laing Q, Xiao W, Dawson R and Ranjan R (2020) FLOOD-PREPARED: A NOWCASTING SYSTEM FOR REAL-TIME IMPACT ADAPTION TO SURFACE WATER FLOODING IN CITIES. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* VI-4-W2-2020: 9–15. DOI:10.5194/isprs-annals-VI-4-W2-2020-9-2020.

Batty M (2007) *Cities and Complexity: Understanding Cities with Cellular Automata, Agent-Based Models, and Fractals*. The MIT Press. ISBN 978-0-262-52479-7.

Blázquez-García A, Conde A, Mori U and Lozano JA (2021) A review on outlier/anomaly detection in time series data. *ACM Computing Surveys (CSUR)* 54(3): 1–33.

Breiman L (2001) Random Forests. *Machine Learning* 45(1): 5–32. DOI:10.1023/A:1010933404324.

Buelvas J, Múnera D, Tobón V DP, Aguirre J and Gaviria N (2023) Data Quality in IoT-Based Air Quality Monitoring Systems: A Systematic Mapping Study. *Water, Air, & Soil Pollution* 234(4): 248. DOI:10.1007/s11270-023-06127-9.

Che Z, Purushotham S, Cho K, Sontag D and Liu Y (2018) Recurrent Neural Networks for Multivariate Time Series with Missing Values. *Scientific Reports* 8(1): 6085. DOI:10.1038/s41598-018-24271-9.

Chen L, Grimstead I, Bell D, Karanka J, Dimond L, James P, Smith L and Edwardes A (2021) Estimating Vehicle and Pedestrian Activity from Town and City Traffic Cameras. *Sensors (Basel)* 21(13): 4564. DOI:10.3390/s21134564.

Cheng H, Feng D, Shi X and Chen C (2018) Data quality analysis and cleaning strategy for wireless sensor networks. *EURASIP Journal on Wireless Communications and Networking* 2018(1): 61. DOI:10.1186/s13638-018-1069-6.

Dembski F, Wössner U, Letzgus M, Ruddat M and Yamu C (2020) Urban Digital Twins for Smart Cities and Citizens: The Case Study of Herrenberg, Germany. *Sustainability* 12(6): 2307. DOI:10.3390/su12062307.

Elkhodr M and Alsinglawi B (2020) Data provenance and trust establishment in the Internet of Things. *Security and Privacy* 3(3): e99.

Gorshenin A, Kozlovskaya A, Gorbunov S and Kochetkova I (2024) Mobile network traffic analysis based on probability-informed machine learning approach. *Computer Networks* 247. DOI:10.1016/j.comnet.2024.110433.

Greff K, Srivastava RK, Koutník J, Steunebrink BR and Schmidhuber J (2017) LSTM: A Search Space Odyssey. *IEEE Transactions on Neural Networks and Learning Systems* 28(10): 2222–2232. DOI:10.1109/TNNLS.2016.2582924.

Gubbi J, Buyya R, Marusic S and Palaniswami M (2013) Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems* 29(7): 1645–1660.

Hastie T, Friedman J and Tibshirani R (2001) Model Inference and Averaging. In: Hastie T, Friedman J and Tibshirani R (eds.) *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York, NY: Springer. ISBN 978-0-387-21606-5, pp. 225–256. DOI:10.1007/978-0-387-21606-5_8.

Hyndman RJ and Koehler AB (2006) Another look at measures of forecast accuracy. *International Journal of Forecasting* 22(4): 679–688. DOI:10.1016/j.ijforecast.2006.03.001.

Karkouch A, Mousannif H, Al Moatassime H and Noel T (2016) Data quality in internet of things: A state-of-the-art survey. *Journal of Network and Computer Applications* 73: 57–81.

Klein A and Lehner W (2009) Representing data quality in sensor data streaming environments. *Journal of Data and Information Quality (JDIQ)* 1(2): 1–28.

Komar T and James P (forthcoming) Multiscale Hierarchical Forecasting of Urban Footfall.

Längkvist M, Karlsson L and Loutfi A (2014) A review of unsupervised feature learning and deep learning for time-series modeling. *Pattern Recognition Letters* 42: 11–24. DOI:10.1016/j.patrec.2014.01.008.

Liu X, Xia Y, Liang Y, Hu J, Wang Y, Bai L, Huang C, Liu Z, Hooi B and Zimmermann R (2023) LargeST: A Benchmark Dataset for Large-Scale Traffic Forecasting. *Advances in Neural Information Processing Systems* 36: 75354–75371.

Martín-Chinea K, Ortega J, Gómez-González JF, Pereda E, Toledo J and Acosta L (2023) Effect of time windows in LSTM networks for EEG-based BCIs. *Cognitive Neurodynamics* 17(2): 385–398. DOI:10.1007/s11571-022-09832-z.

Mohammadi N and Taylor JE (2017) Smart city digital twins. In: *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE. ISBN 1-5386-2726-4, pp. 1–5.

Nguyen HD, Tran KP, Thomassey S and Hamad M (2021) Forecasting and Anomaly Detection approaches using LSTM and LSTM Autoencoder techniques with the applications in supply chain management. *International Journal of Information Management* 57: 102282.

Pascanu R, Gulcehre C, Cho K and Bengio Y (2013) How to Construct Deep Recurrent Neural Networks .

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A and Cournapeau D (????) Scikit-learn: Machine Learning in Python. *MACHINE LEARNING IN PYTHON* .

Press WH and Rybicki GB (1989) Fast algorithm for spectral analysis of unevenly sampled data. *Fast algorithm for spectral analysis of unevenly sampled data* 338(1): 277–280.

Sarrab M, Pulparambil S and Awadalla M (2020) Development of an IoT based real-time traffic monitoring system for city governance. *Global Transitions* 2: 230–245. DOI:10.1016/j.glt.2020.09.004.

Sherstinsky A (2020) Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. *Physica D: Nonlinear Phenomena* 404: 132306. DOI:10.1016/j.physd.2019.132306.

Shukla RM and Sengupta S (2020) Scalable and robust outlier detector using hierarchical clustering and long short-term memory (lstm) neural network for the internet of things. *Internet of Things* 9: 100167.

Şimsek M, Kök İ and Özdemir S (2024) DeepFogAQ: A fog-assisted decentralized air quality prediction and event detection system. *Expert Systems with Applications* 251. DOI:10.1016/j.eswa.2024.123920.

Smith L and Turner M (2019) Building the Urban Observatory: Engineering the largest set of publicly available real-time environmental urban data in the UK. In: *Geophysical Research Abstracts*, volume 21. ISBN 1029-7006.

Teh HY, Kempa-Liehr AW and Wang KIK (2020) Sensor data quality: A systematic review. *Journal of Big Data* 7(1): 11. DOI:10.1186/s40537-020-0285-1.

Van Zoest V, Liu X and Ngai E (2021) Data Quality Evaluation, Outlier Detection and Missing Data Imputation Methods for IoT in Smart Cities. In: Ghosh U, Maleh Y, Alazab M and Pathan ASK (eds.) *Machine Intelligence and Data Analytics for Sustainable Future Smart Cities*. Cham: Springer International Publishing. ISBN 978-3-030-72065-0, pp. 1–18. DOI:10.1007/978-3-030-72065-0_1.

VanderPlas JT (2018) Understanding the lomb–scargle periodogram. *The Astrophysical Journal Supplement Series* 236(1): 16.

Wang RY and Strong DM (1996) Beyond accuracy: What data quality means to data consumers. *Journal of management information systems* 12(4): 5–33.

Yu Y, Si X, Hu C and Zhang J (2019) A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. *Neural Computation* 31(7): 1235–1270. DOI: 10.1162/neco_a_01199.

Zaharia M, Chen A, Davidson A, Ghodsi A, Hong SA, Konwinski A, Murching S, Nykodym T, Ogilvie P, Parkhe M, Xie F and Zumar C (????) Accelerating the Machine Learning Lifecycle with MLflow .