

Diagnosing Seasonality Matters More Than Selecting an Algorithm: A Protocol-Controlled Evaluation of Machine Learning Models for Urban Water Demand Forecasting in Nakuru, Kenya

Dalmas Chituyi Wakhusama
Kabarak University
Nakuru, Kenya

Nelson Masese
Kabarak University
Nakuru, Kenya

Moses M. Thiga
Kabarak University
Nakuru, Kenya

Abstract: The purpose of this study was to evaluate the accuracy and operational suitability of machine learning models for demand-side water forecasting at Nakuru Water and Sanitation Services Company, where 21.7 per cent of the water entering the London Ward network never reached a paying consumer. Four models (ARIMA, FB Prophet, LSTM and Bidirectional LSTM) were re-evaluated on a 964-day operational series under a chronological protocol with a 192-day held-out window, each compared against the naive baseline admissible under its own information set. Models were separated by forecast horizon: lag-based models consume yesterday's realised consumption whereas decomposable models consume only the calendar date, and a rationing schedule must be published in advance. As configured in the parent study, Prophet achieved a mean absolute error of 128.3 m³/day, effectively identical to a straight line fitted to the training data (129.2 m³/day); it had recovered the trend and nothing further. The training partition's autocorrelation showed a dominant 30-day cycle ($r = +0.867$) and negligible weekly structure ($r = +0.203$), the reverse of the configured seasonality. Re-specifying Prophet with a cycle length selected on an inner validation split reduced held-out error to 47.7 m³/day, a 62.8 per cent reduction, outperforming the best lag-based model by 16.9 per cent without observed consumption. The study concluded that diagnostic-driven seasonality specification contributed more to forecast accuracy than algorithm choice, and recommends deriving seasonal structure from the training autocorrelation rather than from software defaults.

Keywords: Demand-side water management; machine learning; seasonality diagnosis; forecast evaluation; non-revenue water; time-series forecasting.

1. INTRODUCTION

Urban water utilities in Kenya operate under chronic supply deficits that force them to ration. Nakuru Water and Sanitation Services Company (NAWASCO) supplies London Ward, Nakuru, on a rotating twelve-hour schedule, and the operational record examined in this study shows why the scheduling problem is acute. Over the 964 days from 1 January 2022 to 21 August 2024, 604,323 m³ of water reached consumers while 167,666 m³ was lost to pipe leakage. Losses therefore accounted for 21.7 per cent of the water put into the network. In national terms that is comparatively good performance: it falls inside the 25 per cent non-revenue water threshold that the Water Services Regulatory Board (WASREB) treats as acceptable, and well below the 48 per cent sector average reported for Kenyan utilities in 2024/25 (WASREB, 2026). The scheduling problem examined here is therefore not an artefact of an unusually weak utility. Even where losses are contained, a fifth of the water entering the network never reaches a paying consumer, and what remains must be rationed across the ward on a timetable that has to be published in advance.

Rationing is not merely an engineering response to scarcity; it is a demand-side management instrument, and its effectiveness depends on being announced in advance. A household that knows when water will flow can store, schedule and economise, which is the mechanism through which perceived behavioural control translates into conservation behaviour (Ajzen, 1991). A household that discovers the supply schedule only when the taps run dry cannot respond. The forecasting requirement that follows is

specific and frequently overlooked: a utility does not need to know what consumption will be tomorrow given what it was today, it needs to know what consumption will be across a horizon of weeks, at the moment the schedule is drafted, when none of the intervening days has been observed.

Machine learning approaches to urban water demand forecasting are well established. Reviews by Donkor, Mazzuchi, Soyer and Roberson (2014) and Ghalekhondabi, Ardjmand, Young and Weckman (2017) document a large literature spanning regression, ARIMA, artificial neural networks and, more recently, recurrent architectures (Guo et al., 2018; Mu, Zheng, Tao, Zhang & Kapelan, 2020). The consistent narrative is that increasingly sophisticated model classes deliver increasingly accurate forecasts.

That narrative rests on evaluation practice that is often too weak to support it. Three problems recur. First, reported errors are frequently not anchored to a naive benchmark, although forecasting methodology has long held that a forecast claim is uninterpretable without one (Hyndman & Koehler, 2006; Hyndman & Athanasopoulos, 2021). Second, in-sample fitted error and out-of-sample forecast error are reported interchangeably, so a reader cannot tell whether a figure represents skill or fit. Third, and least examined, models are compared across different forecast horizons and different information sets, so an apparent ranking may reflect what each model was permitted to see rather than how well it performs.

These weaknesses are not merely academic. If a utility selects a model on the basis of an inflated comparison, it deploys the wrong tool. This study therefore addresses a gap that is both

methodological and practical: there is no published, protocol-controlled comparison of classical, decomposable and recurrent forecasting models for a Kenyan urban water utility in which naive baselines are reported alongside model error and models are separated by the information available to them.

The study pursued two objectives drawn from the parent research (Wakhusama, 2026): to develop a machine learning model for demand-side water forecasting integrating utility operational data, and to evaluate that model's accuracy and operational suitability. It makes three contributions. It reports a protocol-controlled benchmark of four model classes on a Kenyan utility series with horizon-matched naive baselines. It demonstrates that the decomposable model originally selected for deployment had, under its reported configuration, recovered nothing beyond a linear trend. And it shows that correcting the seasonal specification on the basis of a training-data diagnostic reduced forecast error by 62.8 per cent, a margin far larger than any difference between model classes.

2. LITERATURE REVIEW

2.1 Demand-Side Water Management in Supply-Constrained Systems

Demand-side water management shifts the locus of intervention from augmenting supply to shaping consumption, and the applied literature identifies two principal levers. Pricing instruments, of which increasing block tariffs are the most widely adopted form, raise the marginal cost of discretionary use while protecting a lifeline allocation. Information and scheduling instruments (smart metering, consumption feedback and published supply schedules) reduce waste by making consumption legible to both the utility and the consumer, and operate through perceived behavioural control rather than through price (Ajzen, 1991).

Both levers depend on forecasting. A block tariff requires an estimate of the consumption distribution in order to set band boundaries. A published rationing schedule requires an estimate of demand across the scheduling horizon. In each case the forecast must be produced before the period it governs, which constrains the class of models that can be used at all, a constraint that the evaluation literature rarely formalises.

2.2 Machine Learning for Urban Water Demand Forecasting

Classical approaches begin with the Box-Jenkins family (Box, Jenkins, Reinsel & Ljung, 2015). ARIMA models capture short-memory autoregressive and moving-average structure and remain a standard reference in this domain (Adamowski, Fung Chan, Prasher, Ozga-Zielinski & Sliusarieva, 2012). Their limitation for scheduling is structural rather than statistical: an ARIMA forecast at horizon h degenerates towards the series mean as h grows, because it carries no mechanism for calendar-driven variation.

Decomposable additive models address that limitation directly. Prophet represents a series as the sum of a piecewise-linear trend, a set of Fourier seasonal terms and a holiday component (Taylor & Letham, 2018). Because each component is separately inspectable, the model can support a decision that must be explained to a regulator or to the public, a material consideration for a utility justifying rationing. Critically for the present study, Prophet's seasonal terms are user-specified: the software provides weekly and

yearly terms by default, and any other periodicity must be added explicitly by the analyst.

Prophet's own successor addresses its most cited weakness. NeuralProphet (Triebe, Hewamalage, Pilyugina, Laptev, Bergmeir & Rajagopal, 2021) re-implements the framework on PyTorch rather than Stan and adds what its authors term local context: an autoregression module built on AR-Net (Triebe, Laptev & Rajagopal, 2019) and a lagged-covariate module, each configurable either as a linear regression or as a neural network. A single AR-Net produces all h forecast steps at once from the p most recent observations, rather than requiring one fitted model per step. The authors report accuracy gains of 55 to 92 per cent over Prophet on real-world series at short to medium horizons, while retaining separately inspectable trend, seasonality and event components, and the PyTorch backend makes the framework far easier to extend. Two of its properties bear directly on the present study. The reported gain is attributed to the autoregressive terms, so it is realisable only where recent observations exist at the forecast origin. And seasonality in NeuralProphet is still represented by Fourier terms whose periods the analyst supplies, activated by the same weekly and yearly defaults that Prophet applies to daily data. Adopting the newer tool would not, by itself, discharge the diagnostic obligation this paper identifies.

Recurrent neural approaches represent the current frontier. The LSTM cell addresses the vanishing-gradient problem that limited earlier recurrent networks (Hochreiter & Schmidhuber, 1997), and the bidirectional variant processes sequences in both directions (Schuster & Paliwal, 1997). Applications to water demand report strong results at hourly and daily resolution (Herrera, Torgo, Izquierdo & Pérez-García, 2010; Brentan, Luvizotto, Herrera, Izquierdo & Pérez-García, 2017; Guo et al., 2018; Mu et al., 2020). These models are typically evaluated on one-step-ahead tasks in which the previous observation is supplied as an input.

2.3 Evaluation Practice and the Baseline Problem

The methodological literature is unambiguous that forecast accuracy claims require a reference point. Hyndman and Koehler (2006) show that scale-dependent error measures are interpretable only relative to a benchmark, and the M4 competition found that many sophisticated methods failed to outperform simple statistical benchmarks on real series (Makridakis, Spiliotis & Assimakopoulos, 2020). Armstrong (2001) treats benchmark comparison as a precondition of any forecasting claim.

Applied water-utility studies observe these conventions inconsistently. Naive baselines are often absent; the in-sample or out-of-sample provenance of a reported metric is often unstated; and models given different information sets are commonly placed in a single ranked table. A model supplied with yesterday's realised consumption and a model supplied only with a date are performing different tasks, and comparing their errors measures the difference in information rather than the difference in capability.

A related and less-discussed hazard concerns seasonal specification. Software defaults encode assumptions: Prophet's weekly and yearly terms reflect the business and retail series for which the tool was designed. Where the series under study has a different dominant periodicity, accepting the default silently removes the model's principal source of predictive signal. This

study provides a direct empirical demonstration of that failure mode and of its repair.

3. METHODOLOGY

3.1 Study Area and Data

The study used a 964-day daily series covering 1 January 2022 to 21 August 2024, reconstructed from NAWASCO's historical operational records for London Ward, Nakuru. Live smart-meter application programming interface access was subscription-gated at the time of the study, so the series was assembled as a representative proxy for the high-frequency environment the forecasting system is intended to serve. Absolute error magnitudes reported here are therefore protocol-comparable rather than directly transferable to production deployment; the contribution of the study lies in the relative comparisons and in the evaluation method. Table 1 summarises the series and Figure 1 plots it alongside the train/test partition.

Table 1. Characteristics of the NAWASCO London Ward daily series

Property	Value
Observations	964 daily records (one per calendar day)
Period	1 January 2022 to 21 August 2024
Mean daily consumption	626.9 m ³ (SD 169.4)
Range	225.9 to 1,072.7 m ³
Mean absolute deviation about the mean	142.7 m ³
Total consumption	604,323 m ³
Total pipe leakage	167,666 m ³ (21.7% of water input)
Complaints recorded	5,001
Population served	10,000 to 16,131
Zones / supply areas	3 / 6
Hours of supply	12 (constant across all records)

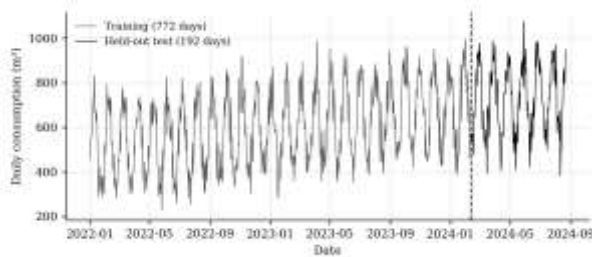


Figure 1. Daily water consumption for London Ward, 1 January 2022 to 21 August 2024, showing the chronological training and held-out test partitions.

Two structural properties constrain interpretation. Hours of supply is invariant, so the volume entering the network is approximately fixed and consumption and leakage are near-complementary by accounting; consistent with this, the two are negatively correlated ($r = -0.405$). Rainfall is uncorrelated with consumption ($r = -0.015$),

indicating that demand in this ward is supply-driven rather than weather-driven.

3.2 Evaluation Protocol

All models were evaluated under one protocol, specified in advance:

1. A single chronological split with no shuffling. The first 772 days (1 January 2022 to 11 February 2024) formed the training partition and the final 192 days (12 February to 21 August 2024) the held-out test window. The split point was fixed at 80 per cent of the available sequences under a seven-day lookback, reproducing the partition used by the parent study's neural models so that their published figures remain comparable.
2. An identical test window and an identical seven-day lookback for every model.
3. Mean absolute error (MAE), root mean square error (RMSE) and mean absolute percentage error (MAPE) computed on the held-out window only, with MAPE defined as the mean of the absolute difference between observed and predicted values divided by the observed value, expressed as a percentage.
4. Naive baselines computed on the same window: persistence (the previous day's value), seasonal persistence at seven and thirty days, the training mean, and a linear trend extrapolated from the training partition.
5. Separation of models by information set. ARIMA, LSTM and Bi-LSTM receive the previous day's realised consumption and produce a day-ahead nowcast. Prophet receives only the calendar date and produces a forecast across the full 192-day horizon from a single origin. These are different tasks and are reported separately.

Four of the five models evaluated in the parent study were re-run under this protocol. The Incremental LSTM was excluded because its online partial-fit update regime continuously ingests new observations and therefore cannot be evaluated against a fixed forecast origin without altering the information set; its published figure was not independently reproduced and is not reported here.

3.3 Model Configurations

ARIMA was specified as (2,1,2) and applied as a rolling one-step-ahead forecast with the model refitted at each step on an expanding window. Prophet was configured as in the parent study, with changepoint prior scale 0.05, yearly and weekly seasonality enabled, daily seasonality disabled, and Kenyan public holidays included. The LSTM comprised two stacked recurrent layers of fifty units with 0.2 dropout and a dense output; the Bi-LSTM used the same topology with bidirectional wrappers. Both were trained with the Adam optimiser on mean squared error for up to 100 epochs with batch size 32, a 10 per cent validation split, and early stopping with patience 10 and best-weight restoration. Inputs were min-max scaled and the random seed was fixed at 42.

3.4 Seasonality Diagnosis and Re-Specification

After the initial benchmark, the autocorrelation function of the training partition was examined. The re-specified Prophet configuration adds a single Fourier seasonal term whose period was selected by an inner validation procedure that never accessed the test window: the model was fitted on training days 0-681 and scored on training days 682-771; candidate periods of 7, 14, 28, 29,

30, 30.4375, 31, 32, 35, 60 and 90 days were swept; the best was chosen on validation error alone; and the selected configuration was then refitted on the full 772-day training partition and scored once on the untouched test window. Fourier order was subjected to the same procedure.

3.5 Scope

This paper evaluates predictive accuracy and operational suitability. Realised water conservation impact is not measured, as that would require a post-deployment control period; it is identified as future work.

4. RESULTS AND DISCUSSION

4.1 The Information Set, not the Algorithm, Determines the Apparent Ranking

Table 2 reports held-out performance separated by forecast task, and Figure 2 presents the same comparison graphically.

Table 2. Model performance by forecast task on the 192-day held-out window. Baselines are shown in italics.

Model / baseline	MAE (m ³ /day)	RMSE (m ³ /day)	MAPE (%)
Panel A: Day-ahead nowcast (previous day's realised consumption)			
Bi-LSTM	57.4	74.2	8.6
LSTM	57.8	74.4	8.6
ARIMA(2,1,2)	63.2	80.5	9.4
<i>Persistence (previous day)</i>	<i>70.0</i>	<i>90.1</i>	<i>10.5</i>
Panel B: Scheduling horizon (calendar date only; 192-day path from one origin)			
Prophet + diagnosed cycle	47.7	60.6	7.2
Prophet, configuration as reported	128.3	150.1	19.2
<i>Linear trend extrapolated from training</i>	<i>129.2</i>	<i>150.3</i>	<i>19.4</i>
<i>Training mean</i>	<i>159.1</i>	<i>194.6</i>	<i>20.4</i>
<i>Seasonal persistence, 7 days</i>	<i>179.0</i>	<i>207.6</i>	<i>26.2</i>
<i>Seasonal persistence, 30 days †</i>	<i>69.0</i>	<i>85.6</i>	<i>10.3</i>

† Requires an observation from 30 days before each target date, so it is admissible only for horizons up to 30 days, whereas the Prophet configurations forecast the full 192-day path from one origin.

Panel A reproduces the expected ordering: the bidirectional recurrent model is the most accurate, and the margin over trivial persistence is 18.0 per cent. This is a modest return for a trained recurrent network, and the explanation lies in the data rather than in the engineering. Lag-1 autocorrelation on the training partition is +0.849. A series that closely resembles its own previous value is largely explained by persistence, so every lag-based model converges to a similar error floor between 57 and 63 m³/day.

Panel B is where the parent study's conclusions required revision, and it also demonstrates why the two panels must not be merged. The lag-based models of Panel A cannot perform the scheduling task at all, because the input they require, yesterday's realised consumption, does not exist at the moment a month of schedule is drafted. Ranking all models in a single table, as the parent study did, measures the information each was given rather than its capability.

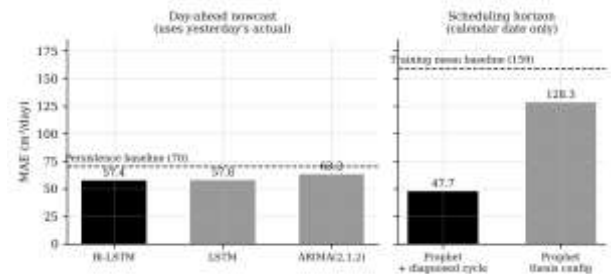


Figure 2. Mean absolute error on the held-out window, grouped by information set. Dashed lines mark the naive baseline admissible within each group.

4.2 The Deployed Configuration had Recovered only a Trend

Under the configuration reported in the parent study, Prophet achieved MAE 128.3, RMSE 150.1 and MAPE 19.2 per cent. A straight line fitted to the training partition and extrapolated across the test window achieved MAE 129.2, RMSE 150.3 and MAPE 19.4 per cent. The two are indistinguishable to within a single m³/day on every metric.

This is a diagnosis, not merely an unfavourable comparison. It establishes that the seasonal component of the fitted model contributed essentially nothing and that the model's predictive content was its linear trend alone. The visual evidence in Figure 4 is unambiguous: the fitted forecast traces a near-flat band close to 700–780 m³, broken only by narrow holiday dips, while the observed series traverses 382 to 1,073 m³ in a regular monthly oscillation that the forecast does not follow at all.

A secondary check reinforces the point. Prophet's error of 128.3 sits close to the series' mean absolute deviation about its own mean, 142.7 m³, which is the score attainable by a model that has learned nothing beyond the central tendency.

4.3 The Series is Governed by a Monthly Cycle, not a Weekly One

The autocorrelation function of the training partition, shown in Figure 3, is a clean damped cosine. It peaks at lag 30 ($r = +0.867$), troughs at lag 15 ($r = -0.617$) and crosses zero between lags 8 and 9 and again between lags 21 and 22. Autocorrelation at lag 7, the weekly lag that the deployed configuration was built to exploit, is only +0.203.

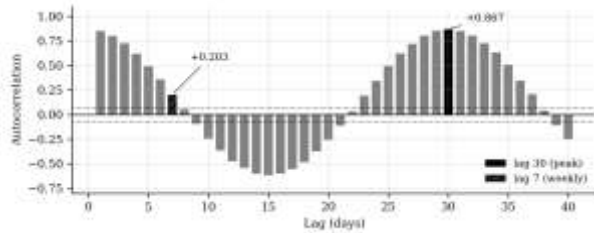


Figure 3. Autocorrelation function of the training partition (772 days), lags 1-40. The dominant cycle is at lag 30 ($r = +0.867$); the weekly lag carries $r = +0.203$. Dotted lines mark the 95 per cent confidence bounds.

Calendar aggregation confirms the reading. Mean consumption by day of week spans 15.5 m^3 against a series standard deviation of 169.4 (under a tenth of one standard deviation), and the highest day is Wednesday, not a weekend day. Mean consumption by day of month, by contrast, spans 283.2 m^3 , peaking in the opening days of the month and at month end at approximately 770 m^3 and falling to approximately 487 m^3 around the eighteenth.

The deployed configuration had therefore been given a weekly seasonal term that carries almost no signal, a yearly term too slow to help across a six-month horizon, and no term at all for the cycle that dominates the series. The consequence, that seven-day seasonal persistence (MAE 179.0) performs worse than simply predicting a constant (MAE 159.1), follows directly and is visible in Table 2.

4.4 Re-Specification Reduced Error by 62.8 Per Cent

Candidate periods were swept on the inner validation split, with results in Table 3.

Table 3. Cycle-length selection on the inner validation window (training days 682-771). No test-window data was used at this stage.

Candidate period (days)	Validation MAE (m^3/day)
None (configuration as reported)	144.6
7	143.7
14	143.8
28	164.9
29	174.2
30	56.1
30.4375	141.4
31	178.8
32	152.7
35	145.9
60	56.1
90 (selected)	54.8

The selection procedure chose 90 days. Because a Fourier expansion of order five over a 90-day period contains a harmonic at exactly 30 days, this configuration represents the monthly cycle with additional flexibility rather than a different structure; a 30-day period selected directly yields a near-identical held-out error of

47.9 against 47.7. Results were also stable across Fourier order, with held-out MAE ranging from 47.7 to 48.7 for orders three through ten.

Refitted on the full training partition and scored once on the untouched test window, the re-specified model achieved **MAE 47.7, RMSE 60.6 and MAPE 7.2 per cent**. Relative to the configuration as reported, this is a **62.8 per cent reduction in mean absolute error**. It also outperforms the 30-day seasonal persistence rule by 30.9 per cent and the best day-ahead recurrent model by 16.9 per cent, despite operating under a strictly poorer information set: the calendar date alone, across a 192-day horizon, from a single forecast origin. Figure 4 contrasts the two configurations against the observed series.

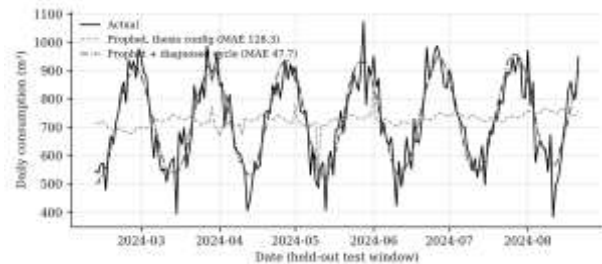


Figure 4. Observed consumption against the two Prophet configurations across the 192-day held-out test window. The configuration reported in the parent study tracks only the trend; the re-specified configuration follows the monthly cycle.

The wider methodological implication is this paper's central claim. The spread across model classes in Panel A is $5.8 \text{ m}^3/\text{day}$, from 57.4 to 63.2. The spread between two configurations of a single model class in Panel B is $80.6 \text{ m}^3/\text{day}$, from 128.3 to 47.7, fourteen times larger. On this series, how the model was specified mattered far more than which model was chosen.

4.5 The Advantage Holds at Every Forecast Distance

A single average over 192 days can conceal a crossover. Table 4 resolves the comparison by forecast distance. Every column forecasts from one origin and sees no observation after the training cut-off, so the information set is identical throughout; the recurrent models are run recursively, each prediction fed back as the next input.

Table 4. Mean absolute error (m^3/day) by forecast distance on the 192-day held-out window. All columns forecast from a single origin with no observed consumption after the training cut-off.

Days ahead	LSTM, recursive	Bi-LSTM, recursive	Prophet, as reported	Prophet + diagnosed cycle
1-7	70.4	79.3	140.4	43.8
8-30	137.4	143.7	152.7	40.5
31-90	127.4	129.3	125.7	45.9
91-192	130.0	130.6	123.6	50.7

Whole window	127.9	129.9	128.3	47.7
--------------	-------	-------	-------	------

Run recursively, the recurrent models lose the advantage Panel A appeared to give them. The LSTM holds 70.4 m³/day over the first week, but its error roughly doubles in the second and settles near 130 thereafter as each prediction is fed back and the error compounds. The bidirectional variant behaves the same way.

The configuration reported in the parent study shows the opposite profile. It is at its worst in the first week, at 140.4, and improves slightly with distance, which is what a model that has recovered only a trend should do: a trend line is no less accurate at day 150 than at day 5.

The re-specified configuration leads in every band, from 43.8 over the first week to 50.7 across the final hundred days. There is no crossover point and no horizon at which a lag-based model, once denied observations, does better. The gain reported above is therefore not an artefact of averaging a long window.

4.6 Why a Decomposable Model Suits this Utility

The case for the decomposable model rests first on a constraint that is operational rather than statistical. NAWASCO publishes its rotation schedule before the period it governs, and at the moment of drafting none of the days being scheduled has been observed. Any model whose input is the previous day's realised consumption is therefore unavailable by construction, which removes the three most accurate models in Panel A from the task the utility actually needs performed. ARIMA can be run to a long horizon but carries no calendar mechanism, so its forecast decays towards the series mean; the recurrent models can be run recursively but then consume their own predictions and accumulate error across 192 steps. Only a model that maps a date to a quantity survives the constraint.

The second consideration is accountability. Rationing allocates a scarce essential service, and a utility licensed under a national regulator must be able to defend how it allocates. Prophet's additive form makes each contribution to a forecast separately visible: how much came from the trend, how much from the monthly cycle, how much from a public holiday. A fifty-unit bidirectional recurrent network returns a number. When a ward disputes its allocation, or a regulator asks why supply was cut on a particular week, the first artefact can be audited and the second cannot.

Third, the model has to survive contact with utility operations. Prophet fits this series in seconds on an ordinary processor with no accelerator, tolerates the gaps and outliers that meter dropouts and burst events routinely produce, absorbs Kenyan public holidays through a single declaration, and exposes changepoints that show an analyst where an infrastructure intervention shifted the trend. It has no epoch budget, learning rate or random seed to manage. The recurrent models required all three, along with the scaling step that, as reported below, was the source of an actual methodological error. Where a utility's analytical capacity is a small team rather than a department, maintenance burden is a selection criterion and not an afterthought.

Each of these would be a reason to accept some accuracy penalty in exchange for a model the utility can operate and defend. The finding reported here is that no such penalty is required. Once the seasonal term matches the series, the decomposable model is

simultaneously the most accountable and the most accurate option tested, and the trade-off the parent study accepted turns out not to exist.

NeuralProphet is the natural successor for this application, for structural rather than fashionable reasons. The two panels of Table 2 exist because every model in this study had to be assigned either a lag-based or a calendar-based information set. NeuralProphet accommodates both within one specification: the autoregression module supplies local context for day-ahead operational control, while the decomposable backbone continues to generate a calendar-driven path when no recent observation exists. A single fitted object could in principle serve both the daily balancing decision and the monthly schedule. Two cautions apply. Its documented advantage derives from the autoregressive terms and is reported at short to medium horizons, so it should not be assumed to transfer to a 192-day path forecast issued from one origin. And because its seasonality is declared exactly as Prophet's is, and defaults the same way, it would have reproduced the misspecification diagnosed here rather than correcting it. NeuralProphet was not evaluated in this study; doing so under the present protocol is the most direct extension of this work.

4.7 Protocol Effects and Robustness

Two further checks were performed, both of which strengthen confidence in the figures above while correcting the parent study's reporting.

The parent study's dashboards computed metrics inconsistently: ARIMA and Prophet reported in-sample fitted residuals while the recurrent models reported held-out error. Recomputing both quantities for the same models gives ARIMA 64.5 in-sample against 63.2 out-of-sample, and Prophet 131.6 against 128.3. On this series the two differ by under three per cent, but the direction and magnitude of the gap are not predictable in advance, which is precisely why a study must declare which it reports rather than pooling whichever figure each tool happens to display.

The recurrent model scripts fitted the min-max scaler on the complete series before partitioning, which leaks test-window information into the scaling parameters. Refitting the scaler on the training partition alone changes LSTM MAE from 57.8 to 57.9 and leaves Bi-LSTM unchanged at 57.4, a difference below 0.2 per cent, because the test window contains no values outside the training range. The methodological error is real and has been corrected; its effect on the reported conclusions is nil.

4.8 Limitations

Three limitations qualify these results. First, the dataset is a reconstructed proxy rather than live telemetry, so absolute error magnitudes should not be read as deployment expectations. Second, the sharpness of the period response in Table 3 (validation error of 56.1 at a period of 30 days against 174.2 at 29 and 178.8 at 31) indicates a close-to-deterministic 30-day component, which is more characteristic of a reconstructed series than of observed municipal consumption. A real deployment must re-run the diagnostic on live data rather than adopting a 30-day term on the strength of this finding. Third, the evaluation covers a single ward and a single 192-day test window; the magnitude of the improvement is specific to this series even though the diagnostic procedure that produced it is general.

None of these qualifications affects the methodological finding, which concerns the procedure rather than the coefficient: the seasonal structure of a series should be established from its autocorrelation function before a model is configured, and the resulting configuration should be validated without reference to the test data.

5. CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

This study evaluated four machine learning models for demand-side water forecasting at NAWASCO under a single controlled protocol with horizon-matched naive baselines. On the day-ahead task, the bidirectional recurrent model improved on trivial persistence by 18.0 per cent, a modest margin attributable to strong short-lag autocorrelation in the series rather than to architectural sophistication. On the scheduling task, the only task capable of producing a schedule that can be published in advance, the decomposable model as originally configured performed no better than a straight line, having been supplied with a weekly seasonal term the series does not exhibit and no term for the 30-day cycle that governs it. Re-specifying that single component, using a period identified from the training data and selected on an inner validation split, reduced held-out error by 62.8 per cent to a mean absolute error of 47.7 m³/day and a mean absolute percentage error of 7.2 per cent, outperforming every lag-based model while requiring no observed consumption. The difference between two configurations of one model exceeded the difference between model classes by a factor of fourteen. The study concludes that on series of this character, diagnostic-driven specification contributes more to forecast accuracy than algorithm selection, and that a correctly specified decomposable model delivers both the accuracy and the transparency a regulated utility requires.

5.2 Recommendations

First, utility forecasting studies should establish seasonal structure from the autocorrelation function of the training partition before configuring a model, rather than accepting software defaults, and should report the diagnostic alongside the result. Second, such studies should declare the forecast horizon and information set, report the naive baseline admissible under that information set, state whether errors are in-sample or out-of-sample, and refrain from ranking models that were given different information. Third, NAWASCO should adopt the re-specified configuration for schedule generation, record zone and time-slot assignment against operational conditions rather than by rotation so that allocation decisions become learnable, and re-run the seasonality diagnostic once live smart-meter telemetry becomes available. Fourth, future research should evaluate NeuralProphet under this protocol, since it is the one framework tested or reviewed here that can serve the day-ahead and scheduling tasks from a single fitted model, and should measure realised conservation impact over a post-deployment control period, which the present study could not do. Whether the dominance of specification over architecture reported here holds across other Kenyan water service providers also remains to be established.

6. ACKNOWLEDGMENTS

The authors thank Nakuru Water and Sanitation Services Company (NAWASCO) for making the operational records for London Ward available for this study, and colleagues at Kabarak University for feedback on earlier drafts.

7. REFERENCES

- [1] Adamowski, J., Fung Chan, H., Prasher, S. O., Ozga-Zielinski, B., & Sliusarieva, A. (2012). Comparison of multiple linear and nonlinear regression, autoregressive integrated moving average, artificial neural network, and wavelet artificial neural network methods for urban water demand forecasting in Montreal, Canada. *Water Resources Research*, 48(1), W01528.
- [2] Ajzen, I. (1991). The theory of planned behavior. *Organizational Behavior and Human Decision Processes*, 50(2), 179-211.
- [3] Armstrong, J. S. (Ed.). (2001). *Principles of forecasting: A handbook for researchers and practitioners*. Kluwer Academic Publishers.
- [4] Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time series analysis: Forecasting and control* (5th ed.). John Wiley & Sons.
- [5] Brentan, B. M., Luvizotto Jr, E., Herrera, M., Izquierdo, J., & Pérez-García, R. (2017). Hybrid regression model for near real-time urban water demand forecasting. *Journal of Computational and Applied Mathematics*, 309, 532-541.
- [6] Donkor, E. A., Mazzuchi, T. A., Soyer, R., & Roberson, J. A. (2014). Urban water demand forecasting: Review of methods and models. *Journal of Water Resources Planning and Management*, 140(2), 146-159.
- [7] Ghalekhondabi, I., Ardjmand, E., Young, W. A., & Weckman, G. R. (2017). Water demand forecasting: Review of soft computing methods. *Environmental Monitoring and Assessment*, 189(7), 313.
- [8] Guo, G., Liu, S., Wu, Y., Li, J., Zhou, R., & Zhu, X. (2018). Short-term water demand forecast based on deep learning method. *Journal of Water Resources Planning and Management*, 144(12), 04018076.
- [9] Herrera, M., Torgo, L., Izquierdo, J., & Pérez-García, R. (2010). Predictive models for forecasting hourly urban water demand. *Journal of Hydrology*, 387(1-2), 141-150.
- [10] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735-1780.
- [11] Hyndman, R. J., & Athanasopoulos, G. (2021). *Forecasting: Principles and practice* (3rd ed.). OTexts.
- [12] Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4), 679-688.
- [13] Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2020). The M4 Competition: 100,000 time series and 61 forecasting methods. *International Journal of Forecasting*, 36(1), 54-74.
- [14] Mu, L., Zheng, F., Tao, R., Zhang, Q., & Kapelan, Z. (2020). Hourly and daily urban water demand predictions using a long short-term memory based model. *Journal of Water Resources Planning and Management*, 146(9), 05020017.

- [15] Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11), 2673-2681.
- [16] Taylor, S. J., & Letham, B. (2018). Forecasting at scale. *The American Statistician*, 72(1), 37-45.
- [17] Triebe, O., Laptev, N., & Rajagopal, R. (2019). AR-Net: A simple auto-regressive neural network for time-series. *arXiv*. <https://doi.org/10.48550/arXiv.1911.12436>
- [18] Triebe, O., Hewamalage, H., Pilyugina, P., Laptev, N., Bergmeir, C., & Rajagopal, R. (2021). NeuralProphet: Explainable forecasting at scale. *arXiv*. <https://doi.org/10.48550/arXiv.2111.15397>
- [19] Wakhusama, D. C. (2026). A model for enhancing demand-side water management through machine learning and IoT integration [Unpublished master's thesis]. Kabarak University.
- [20] Water Services Regulatory Board. (2026). Impact: A performance report of Kenya's water services sector 2024/25 (Issue No. 18). WASREB.