Forecasting School Meal Production Costs: A Comparative Study of Machine Learning and Deep Learning Time-Series Models

Chaya Chandana Doddaiggaluru Appajigowda*, Varshith Reddy Bhimireddy*, Areena*, Amir Jafari†
*Graduate Students, M.S. in Data Science, The George Washington University, Washington, D.C.
Emails: chayachandanad@gwu.edu, varshithreddy.bhimireddy@gwu.edu, Areenas@gwu.edu
†Assistant Professor, Data Science Program, The George Washington University, Washington, D.C.
Email: ajafari@gwu.edu

***Abstract*—Planning meal production in Fairfax County Public Schools (FCPS) is challenging due to highly irregular student participation and substantial food waste. Traditional forecasting methods struggle to model these nonlinear and interdependent patterns. To address this, we developed and evaluated a suite of machine learning and deep learning models including Linear Regression, XGBoost, Feedforward Neural Networks (FNN), Gated Recurrent Units (GRU), and Long Short-Term Memory (LSTM) networks using real FCPS multivariate meal production data. Among all models, the LSTM achieved the best performance, reducing the mean forecasting error from approximately 197 to 137, representing a 30.6% improvement. These findings highlight the potential of deep learning approaches to significantly enhance production cost forecasting, reduce waste, support budgeting accuracy, and improve data-driven decision-making for school meal programs.**

I. Introduction

Accurately forecasting daily production costs in school meal programs presents ongoing challenges for district administrators, nutrition service directors, and operational planners. Despite improvements in digital record-keeping and menu standardization, school districts continue to experience substantial fluctuations in student participation, ingredient prices, and food waste. These variations introduce instability in budget allocation, complicate daily meal planning, and increase the risk of operational inefficiencies. Traditional forecasting approaches such as heuristic rules, fixed multipliers, or linear extrapolation-lack the capacity to capture the nonlinear, irregular, and time-dependent patterns present in real production-cost data, especially when influenced by school-specific behavior, holiday schedules, and supply-chain variability.

Recent advances in machine learning and deep learning offer more robust alternatives by enabling models to learn complex temporal dependencies directly from historical data. In this study, we develop an integrated forecasting framework to predict short-term production costs using historical data aggregated from multiple schools and meal types. The dataset used in this work, meals_combined.csv, was generated by scraping more than 100 HTML files and merging them into a single structured dataset. This process introduced numerous data-quality challenges, including inconsistent currency formatting, ambiguous date structures, missing or irregular entries, and extreme outliers such as isolated days where a school recorded unrealistically high production costs. To address these issues, we constructed a preprocessing pipeline that performs currency coercion, outlier removal, automated date parsing, temporal aggregation, and sliding-window sequence generation, making the dataset suitable for both machine-learning and deep-learning models.

Using this cleaned dataset, we evaluate forecasting performance in both univariate and multivariate settings across a range of models, including Linear Regression, XGBoost, Feed-Forward Neural Networks (FNN), Gated Recurrent Units (GRU), and Long Short-Term Memory (LSTM) networks. All models are trained using a consistent 70%–30% train–test split and evaluated using three standard error-based metrics: Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and the coefficient of determination (R^2). Model performance is further examined through diagnostic visualizations such as predicted-versus-actual plots and multi-day future forecasts, enabling deeper insight into each model's ability to capture temporal structure and cost variability across schools and meal types.

The goal of this work is to provide a comprehensive and comparative analysis of modern forecasting techniques for school meal production-cost prediction while offering a reliable framework that can be deployed by districts to improve budget planning, reduce food waste, and enhance operational efficiency. The remainder of this paper is organized as follows: Section 2 describes the dataset and preprocessing components; Section 3 presents exploratory data analysis; Section 4 details the methodological framework and model architectures; Section 5 reports forecasting results across all models; Section 6 discusses implications for real-world deployment; and Section 7 concludes with limitations and future research directions. Through this investigation, we aim to demonstrate the value of data-driven forecasting approaches in supporting more sustainable and cost-efficient meal operations within K–12 school districts.

II. Related work

Research in school meal management can be categorized into two primary domains: studies related to cafeteria food waste and studies involving forecasting models. Both areas help contextualize the problem addressed in this project.

Food Waste in School Cafeterias

Several studies have reported that schools waste a substantial amount of food, resulting in financial losses and environmental

impacts. Cohen et al. (2013) studied middle school cafeterias and found that students wasted approximately 20% of their meals, with milk and vegetables being the most frequently discarded items. They estimated that each wasted lunch cost roughly $0.53, a figure that accumulates significantly across large school districts.

Byker et al. (2014) observed that after the introduction of healthier meals featuring fruits and vegetables, food waste increased by 56%. This occurred because students were required to take certain items but did not always consume them, highlighting a mismatch between planned menus and actual student consumption.

Smith and Cunningham-Sabo (2014) reported that waste rates vary greatly by menu item. Some meals exhibited waste rates exceeding 35%, whereas more popular meals produced less than 15% waste. They also found that schools often over-prepare food "just in case," contributing to increased waste and higher costs.

These studies show that schools waste both food and financial resources due to inaccurate predictions of required food quantities. This motivates the development of forecasting approaches that can better predict production costs and reduce waste.

Forecasting Models and Deep Learning

Deep learning models have become powerful tools for time-series forecasting. Hochreiter and Schmidhuber (1997) introduced Long Short-Term Memory (LSTM) networks, which addressed limitations in earlier recurrent neural networks by enabling the retention of long-term information through specialized gating mechanisms. LSTMs are particularly effective at capturing repeating patterns over extended time periods.

Cho et al. (2014) proposed the Gated Recurrent Unit (GRU), a simpler and computationally faster alternative to LSTM. Chung et al. (2014) found that LSTMs typically perform better for long sequences, while GRUs are more efficient and perform comparably well on shorter-term forecasts such as the 7-day predictions considered in this study.

These models have been successfully applied in various domains. Kong et al. (2019) used LSTM networks to forecast residential electricity consumption and achieved a prediction error of only 2.28%, outperforming traditional approaches that exhibited 5.73% error. Lai et al. (2018) applied LSTMs to predict retail product demand and reported accuracy improvements of 15–20% over earlier methods.

Despite advancements in forecasting, no prior work has applied deep learning to predict school meal production costs. Existing school-related studies focus primarily on post-meal waste measurement rather than forecasting costs before production. Meanwhile, forecasting research has largely focused on sectors such as energy, retail, and healthcare rather than school nutrition programs.

III. Dataset

This study utilized deidentified, publicly accessible daily meal-production records from Fairfax County Public Schools (FCPS). The original data were not provided in a structured format; instead, they consisted of hundreds of HTML production reports for breakfast and lunch across more than 100 schools. Each HTML file contained item-level tables documenting planned meals, meals served, leftover quantities, discarded items, and total production costs. These variables form the foundation of the forecasting features summarized in Table I.

To convert these unstructured HTML files into an analysis-ready dataset, a custom parsing pipeline was implemented. This pipeline automatically detected school headers, extracted item tables, removed subtotal rows, standardized column names, and cleaned numeric fields. Currency strings such as "$13.32" were transformed into numerical values (13.32), commas and symbols were removed, and percentages were converted to floating-point values. Missing entries were corrected using forward- and backward-fill methods to preserve temporal continuity, and invalid rows were removed entirely.

Outlier detection played a crucial role during preprocessing. A 99th-percentile threshold was applied to the `production_cost_total` column to eliminate extreme or erroneous values. One memorable discovery revealed during early exploratory analysis-highlighted the importance of this step. While reviewing cost distributions with a faculty advisor, an unusually large value appeared: a production cost of approximately **$14,010** for a single item, "*Fat Free Chocolate Milk*," at Hughes Middle School on **16-May-2025**. At first, it seemed so implausible that we assumed our code must be broken or combining costs incorrectly. However, after manually searching the raw dataset, we confirmed the value was genuinely present for that one day and that one menu item. Although humorous in hindsight-imagining a school spending five figures on chocolate milk-the anomaly underscored the necessity of rigorous outlier removal to ensure model stability and faithful forecasting.

After all preprocessing steps were completed, breakfast and lunch datasets were merged into a unified file, `meals_combined.csv`. The resulting cleaned dataset contains approximately **177,492 rows** and about **20 columns**, representing more than 100 FCPS schools and two meal types (breakfast and lunch), covering records through May 2025. These features, detailed in Table I, collectively capture the operational, behavioral, and financial dynamics of FCPS meal production. A smaller subset of these columns, shown in Table II, was used for training univariate and multivariate forecasting models.

These fields provided the basis for all machine-learning and deep-learning forecasting models, enabling the capture of temporal patterns, operational variability, and cost behavior across the FCPS meal production system.

Exploratory Data Analysis (EDA)

A comprehensive exploratory data analysis (EDA) was performed to understand the statistical properties of the dataset, assess data quality, verify distributional behavior of key variables, and evaluate potential multicollinearity among predictors. The analysis focused primarily on the production_cost_total variable, operational meal counts, and the structural relation-

TABLE I
DATA DICTIONARY OF INPUT FEATURES

Name of Feature	Description of Feature
school_name	Name of the FCPS school associated with the production report.
date	Calendar date of the meal production record (YYYY-MM-DD).
identifier	Unique internal identifier for each item entry.
name	Name of the meal item (e.g., entrée, milk, side dish).
planned_reimbursable	Number of planned reimbursable meals.
planned_non_reimbursable	Number of planned non-reimbursable meals.
planned_total	Total number of meals planned.
offered_total	Total quantity of items offered.
served_reimbursable	Count of reimbursable meals served.
served_non_reimbursable	Count of non-reimbursable meals served.
served_total	Total meals served.
served_cost	Cost associated with served items.
discarded_total	Number of meals discarded.
discarded_percent_of_offered	Percentage of offered items discarded.
discarded_cost	Cost of discarded meals.
subtotal_cost	Intermediate cost measure.
left_over_total	Number of leftover meals.
left_over_percent_of_offered	Percentage of offered items left over.
left_over_cost	Cost associated with leftover items.
production_cost_total	Total production cost (target variable).
meal_type	Meal category (Breakfast or Lunch).

TABLE II
COLUMNS USED FOR FORECASTING MODELS

Column	Description
date	Time index for forecasting sequences.
school_name	Grouping variable maintaining school-specific continuity.
meal_type	Breakfast or lunch identifier.
served_total	Lagged feature capturing consumption behavior.
planned_total	Planned demand indicator.
discarded_total	Waste-related signal contributing to variability.
left_over_total	Operational efficiency indicator.
production_cost_total	**Primary target variable** for forecasting tasks.

ships among these features. All analyses were conducted after the dataset was fully cleaned and standardized.

Outlier Detection

Outlier inspection revealed that the production_cost_total variable exhibited a heavy right-skewed distribution, driven by rare but extreme values.

Outliers were identified using the Interquartile Range (IQR) criterion, where a value x is considered an outlier if

$$x < Q1 - 1.5 \times IQR \quad \text{or} \quad x > Q3 + 1.5 \times IQR.$$

Values exceeding the 99th percentile threshold were excluded to stabilize the distribution, reduce model distortion, and improve downstream forecasting performance.

Correlation Analysis

To examine linear relationships among key operational and cost variables, a Pearson correlation matrix was computed and visualized as a heatmap (Figure 1). The matrix highlights strong associations between served_total and production_cost_total, reflecting the expected cost dependence on meal volume. Other predictors demonstrated weak or moderate correlations, suggesting minimal risk of redundancy.

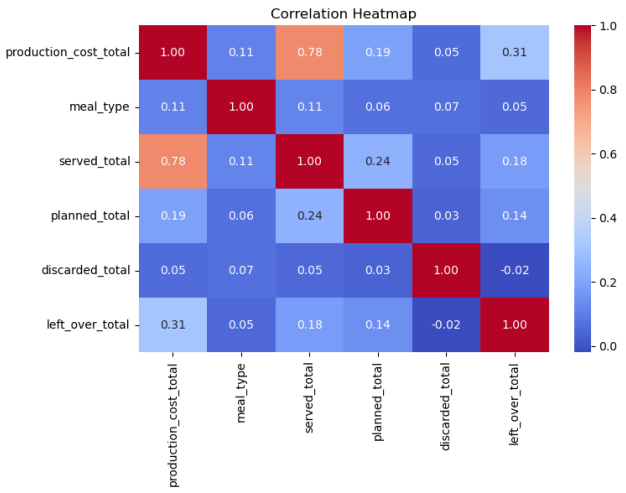

Fig. 1. Correlation heatmap showing pairwise Pearson correlations among numerical features.

Multicollinearity Diagnostics

Multicollinearity among predictors was evaluated using both Variance Inflation Factor (VIF) analysis and Singular Value Decomposition (SVD).

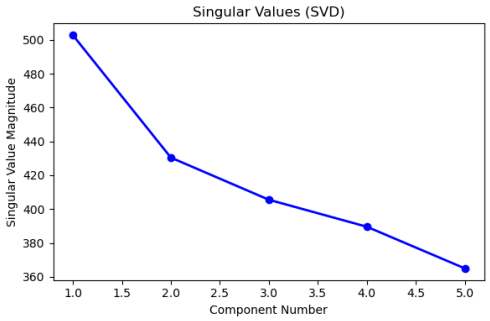

Fig. 2. Singular value magnitudes from the SVD of the standardized feature matrix.

Variance Inflation Factor (VIF): For each predictor X_i, the VIF is defined as:

$$\mathrm{VIF}(X_i) = \frac{1}{1 - R_i^2},$$

where R_i^2 is obtained by regressing X_i on all remaining predictors. All computed VIF values ranged between 1.02 and 1.28, indicating negligible multicollinearity.

Singular Value Decomposition (SVD): To further verify the absence of near-linear dependence, the standardized feature matrix was decomposed using:

$$X = U\Sigma V^T.$$

The resulting singular values (Figure 2) were all substantially greater than zero and gradually decreasing, confirming that the feature space does not suffer from rank deficiency.

Overall, the EDA demonstrated that the dataset is well-conditioned for downstream modeling, with properly handled outliers and no significant redundancy among predictors.

IV. METHODOLOGY

The methodological approach for this study focused on forecasting the daily production_cost_total for each school and meal type. After preparing the dataset through currency cleaning, missing-value handling, and outlier removal using the IQR method, individual time series were constructed for every school–meal combination. Although both multivariate and univariate forecasting models were initially explored, the univariate approach ultimately became the primary method because it consistently delivered higher stability and accuracy. Each time series was transformed into a supervised learning structure using a fixed sliding window, allowing models to learn temporal patterns directly from past production costs.

Multiple forecasting techniques were evaluated, including Linear Regression, XGBoost, Feedforward Neural Networks (FNN), Gated Recurrent Units (GRU), and Long Short-Term Memory (LSTM) networks. Among these, the univariate LSTM demonstrated the strongest performance, effectively capturing nonlinear temporal dependencies in the cost patterns. Model performance was assessed using standard metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), and the coefficient of determination (R^2). Finally, a rolling backtesting procedure was applied on the test set to evaluate real-world forecasting reliability, where the univariate LSTM achieved the highest R^2, confirming its suitability as the final forecasting model.

A. Data Preprocessing

The raw dataset contained daily records for multiple schools and meal types, including variables such as production_cost_total, served_total, planned_total, discarded_total, and leftover_total. Before model development, several preprocessing steps were performed to ensure data consistency and reliability. First, currency fields stored as strings were cleaned and converted into numerical values. Missing entries were addressed using forward filling for short gaps and removal for incomplete rows that could not be reconstructed logically. Outliers in production cost were detected using the Interquartile Range (IQR) method and removed to prevent distortion of temporal learning patterns.

Since each school operates independently, the dataset was partitioned into separate school–meal time series. Dates were standardized to a continuous calendar index, ensuring that weekends and holidays did not produce unintended gaps. For each series, data were scaled using Min–Max normalization fitted only on the training split to avoid leakage. Finally, the cleaned and normalized series were converted into supervised learning format using a fixed sliding window, enabling the forecasting models to learn temporal dependencies directly from historical production cost patterns.

B. pipeline

The forecasting workflow followed a structured pipeline that transformed the raw dataset into model-ready sequences and ultimately produced evaluation metrics. The process began by loading the cleaned dataset and splitting it chronologically into training, validation, and test subsets. The training and validation portions were then passed through a preprocessing module responsible for scaling, sliding-window transformation, and sequence formatting.

After preprocessing, an optional oversampling step was applied to the training data to increase the number of available sequences for models requiring larger sample sizes. The processed data were then fed into the forecasting model, which was trained using the validation set for hyperparameter tuning and early stopping. Once training was complete, the final model generated predictions on the untouched test set. These predictions were evaluated using standard forecasting metrics, forming the basis for model comparison and performance assessment.

Figure 3 provides a schematic overview of this end-to-end pipeline.

C. models

In addition to the univariate forecasting framework used for deployment, we first explored a series of multivariate forecasting models to evaluate whether incorporating additional operational variables (e.g., served total, planned total, discarded total, leftover total) would improve predictive accuracy. These

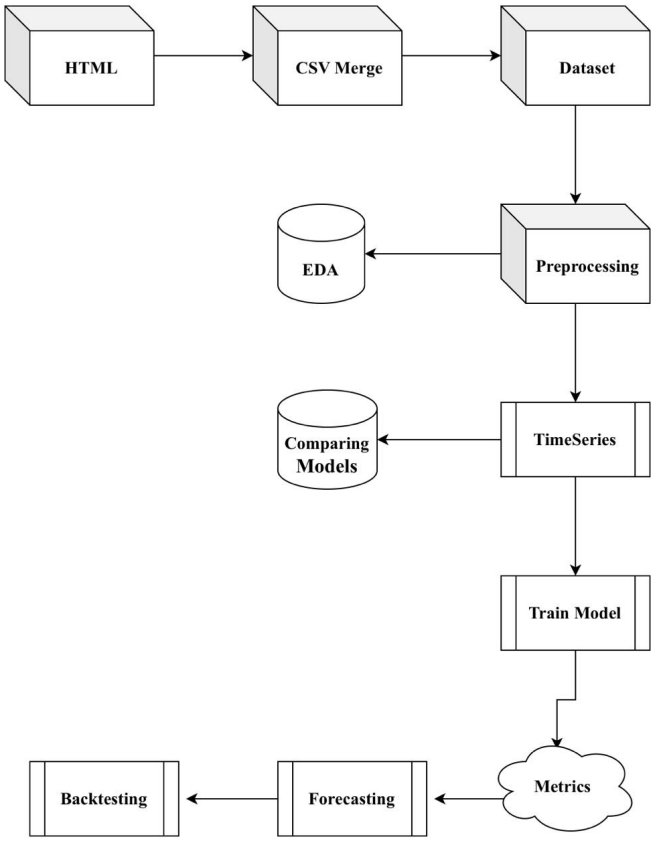

Fig. 3. Time series and forecasting pipeline.

models treat each daily observation as a vector rather than a single scalar production cost.

Let

$$\mathbf{x}_t = \begin{bmatrix} \text{served\_total}_t \\ \text{planned\_total}_t \\ \text{discarded\_total}_t \\ \text{leftover\_total}_t \\ \text{production\_cost\_total}_t \end{bmatrix} \in \mathbb{R}^d,$$

denote the feature vector for day t, where $d = 5$ in our dataset.

Given a window length W, the forecasting task is defined as:

$$\hat{y}_{t+1} = f(\mathbf{x}_t, \mathbf{x}_{t-1}, \ldots, \mathbf{x}_{t-W+1}).$$

The multivariate models were evaluated for comparison; however, the univariate models ultimately offered better stability and were therefore used in final forecasting experiments.

1) Linear Regression (Multivariate): For multivariate linear regression, the concatenated input window is flattened into a single feature vector:

$$\mathbf{z}_t = \left[\mathbf{x}_t^\top, \mathbf{x}_{t-1}^\top, \ldots, \mathbf{x}_{t-W+1}^\top\right]^\top \in \mathbb{R}^{dW}.$$

The model estimates:

$$\hat{y}_{t+1} = \beta_0 + \beta^\top \mathbf{z}_t,$$

where $\beta \in \mathbb{R}^{dW}$ contains the regression coefficients.

Parameters are obtained by solving:

$$\min_{\beta_0,\beta} \sum_{t \in T_{\text{train}}} (y_{t+1} - \hat{y}_{t+1})^2.$$

2) XGBoost Regressor (Multivariate): The multivariate XGBoost model uses the same flattened vector $\mathbf{z}_t$, with prediction:

$$\hat{y}_{t+1} = \sum_{k=1}^{K} f_k(\mathbf{z}_t), \qquad f_k \in \mathscr{F},$$

where each f_k is a regression tree.

The objective is:

$$\mathscr{L} = \sum_{t \in T_{\text{train}}} \ell(y_{t+1}, \hat{y}_{t+1}) + \sum_{k=1}^{K} \Omega(f_k),$$

with regularization:

$$\Omega(f_k) = \gamma T_k + \frac{1}{2}\lambda \sum_{j=1}^{T_k} w_{k,j}^2.$$

3) Feedforward Neural Network (Multivariate): The FNN also receives the flattened feature vector $\mathbf{z}_t$. The first hidden layer computes:

$$\mathbf{h}^{(1)} = \phi\left(W^{(1)}\mathbf{z}_t + \mathbf{b}^{(1)}\right),$$

where $\phi(\cdot)$ is typically ReLU.

Subsequent layers follow:

$$\mathbf{h}^{(\ell)} = \phi\left(W^{(\ell)}\mathbf{h}^{(\ell-1)} + \mathbf{b}^{(\ell)}\right), \quad \ell = 2, \ldots, L.$$

The output layer produces:

$$\hat{y}_{t+1} = \mathbf{w}_{\text{out}}^\top \mathbf{h}^{(L)} + b_{\text{out}}.$$

4) Long Short-Term Memory Network (Multivariate): Multivariate LSTM models treat each day's vector $\mathbf{x}_\tau$ as a time-step input. The LSTM gates are computed as follows:

$$f_\tau = \sigma\left(W_f \begin{bmatrix} h_{\tau-1} \\ \mathbf{x}_\tau \end{bmatrix} + b_f\right),$$

$$i_\tau = \sigma\left(W_i \begin{bmatrix} h_{\tau-1} \\ \mathbf{x}_\tau \end{bmatrix} + b_i\right),$$

$$\tilde{c}_\tau = \tanh\left(W_c \begin{bmatrix} h_{\tau-1} \\ \mathbf{x}_\tau \end{bmatrix} + b_c\right),$$

$$c_\tau = f_\tau \odot c_{\tau-1} + i_\tau \odot \tilde{c}_\tau,$$

$$o_\tau = \sigma\left(W_o \begin{bmatrix} h_{\tau-1} \\ \mathbf{x}_\tau \end{bmatrix} + b_o\right),$$

$$h_\tau = o_\tau \odot \tanh(c_\tau).$$

After processing the full sequence window, the next-day forecast is:

$$\hat{y}_{t+1} = w_{\text{out}}^\top h_t + b_{\text{out}}.$$

5) Gated Recurrent Unit (Multivariate): The GRU computes:

$$z_\tau = \sigma\left(W_z \begin{bmatrix} h_{\tau-1} \\ \mathbf{x}_\tau \end{bmatrix} + b_z\right),$$

$$r_\tau = \sigma\left(W_r \begin{bmatrix} h_{\tau-1} \\ \mathbf{x}_\tau \end{bmatrix} + b_r\right),$$

$$\tilde{h}_\tau = \tanh\left(W_h \begin{bmatrix} r_\tau \odot h_{\tau-1} \\ \mathbf{x}_\tau \end{bmatrix} + b_h\right),$$

$$h_\tau = (1 - z_\tau) \odot h_{\tau-1} + z_\tau \odot \tilde{h}_\tau.$$

Finally:

$$\hat{y}_{t+1} = w_{\text{out}}^\top h_t + b_{\text{out}}.$$

D. Univariate Time-Series Modeling for School-Level Production Costs

In this component of the Smart School Food Service Analytics project, we develop univariate forecasting models to predict daily production costs for each school and meal type. Let $s \in \mathscr{S}$ denote a school and $m \in \{\text{Breakfast}, \text{Lunch}\}$ denote a meal type. For every (s,m) pair, we construct an individual time series $\{y_t^{(s,m)}\}_{t=1}^{T_{s,m}}$, where $y_t^{(s,m)}$ represents the total production cost for school s and meal type m on calendar day t.

The forecasting task is to predict the next-day cost using only its own recent history. Given a window size W, we use the previous W days of $y_t^{(s,m)}$ as input and forecast the cost on the next day:

$$\hat{y}_{t+1}^{(s,m)} = f\left(y_t^{(s,m)}, y_{t-1}^{(s,m)}, \ldots, y_{t-W+1}^{(s,m)}\right), \quad t = W, \ldots, T_{s,m} - 1.$$

In our experiments, we set $W = 3$, which captures short-term temporal dynamics while keeping the input dimensionality low for smaller school series.

We train a separate model for each (s,m) time series, using the same architecture and hyperparameters across schools. The primary model is a Long Short-Term Memory (LSTM) network, and we use Gated Recurrent Units (GRU), Feedforward Neural Networks (FNN), Linear Regression, and XGBoost as benchmark models. This design allows us to quantify the benefit of sequence-aware deep learning models over classical regression and tree-based methods.

1) Data Aggregation and Windowing: The raw dataset contains transaction-level records with columns including `school_name`, `date`, `meal_type`, and `production_cost_total`, among others. For each school s and meal type m, we aggregate all records on a given date t:

$$y_t^{(s,m)} = \sum_{i \in \mathscr{I}(s,m,t)} \texttt{production\_cost\_total}^{(i)},$$

where $\mathscr{I}(s,m,t)$ is the index set of rows corresponding to school s, meal m, and day t.

Each resulting series is then transformed into a supervised learning dataset by forming overlapping input–output pairs using a sliding window of length W. For each $t \geq W$, the input vector and target are defined as:

$$\mathbf{x}_t^{(s,m)} = \begin{bmatrix} y_{t-W+1}^{(s,m)} & \cdots & y_{t-1}^{(s,m)} & y_t^{(s,m)} \end{bmatrix}^\top \in \mathbb{R}^W,$$
$$z_{t+1}^{(s,m)} = y_{t+1}^{(s,m)}.$$

For each (s,m) series, we split the data chronologically into training, validation, and test sets (e.g., earliest $\approx 60\%$ for training, middle $\approx 20\%$ for validation, and most recent $\approx 20\%$ for testing), ensuring that no future information leaks into the past.

2) Scaling and Preprocessing: Because different schools have different cost scales, we normalize each series independently. Let $\mathscr{T}_{\text{train}}^{(s,m)}$ denote the set of training indices for series (s,m). We fit a Min–Max scaler on the training targets:

$$\tilde{y}_t^{(s,m)} = \frac{y_t^{(s,m)} - \min_{k \in \mathscr{T}_{\text{train}}^{(s,m)}} y_k^{(s,m)}}{\max_{k \in \mathscr{T}_{\text{train}}^{(s,m)}} y_k^{(s,m)} - \min_{k \in \mathscr{T}_{\text{train}}^{(s,m)}} y_k^{(s,m)}}, \quad t \in \mathscr{T}_{\text{train}}^{(s,m)}.$$

The same scaling parameters (min and max) are then applied to the validation and test points of the same series. All models are trained on the scaled values $\tilde{y}_t^{(s,m)}$ and their inputs, and predictions are inverse-transformed back to the original currency units for evaluation.

E. Forecasting Models

For each (s,m) series, we train one instance of each model described below. Hyperparameters (e.g., learning rate, number of epochs, tree depth) are tuned on the validation set. The final comparison is made on the held-out test set.

1) Long Short-Term Memory (LSTM): Long Short-Term Memory (LSTM) networks [3] are a class of recurrent neural networks specifically designed to capture long- and short-term dependencies in sequential data. For a given series (s,m), we feed the sequence $\{\tilde{y}_t^{(s,m)}\}$ into an LSTM with L stacked layers and hidden state dimension H.

At each time step τ in the window, the LSTM cell receives the current input x_τ (scaled cost at time τ) and the previous hidden and cell states $(\mathbf{h}_{\tau-1}, \mathbf{c}_{\tau-1})$. The cell computes:

$$\begin{aligned}
\mathbf{f}_\tau &= \sigma\left(\mathbf{W}_f \begin{bmatrix} \mathbf{h}_{\tau-1} \\ x_\tau \end{bmatrix} + \mathbf{b}_f\right), \\
\mathbf{i}_\tau &= \sigma\left(\mathbf{W}_i \begin{bmatrix} \mathbf{h}_{\tau-1} \\ x_\tau \end{bmatrix} + \mathbf{b}_i\right), \\
\tilde{\mathbf{c}}_\tau &= \tanh\left(\mathbf{W}_c \begin{bmatrix} \mathbf{h}_{\tau-1} \\ x_\tau \end{bmatrix} + \mathbf{b}_c\right), \\
\mathbf{c}_\tau &= \mathbf{f}_\tau \odot \mathbf{c}_{\tau-1} + \mathbf{i}_\tau \odot \tilde{\mathbf{c}}_\tau, \\
\mathbf{o}_\tau &= \sigma\left(\mathbf{W}_o \begin{bmatrix} \mathbf{h}_{\tau-1} \\ x_\tau \end{bmatrix} + \mathbf{b}_o\right), \\
\mathbf{h}_\tau &= \mathbf{o}_\tau \odot \tanh(\mathbf{c}_\tau),
\end{aligned}$$

where $\sigma(\cdot)$ is the logistic sigmoid, and $\odot$ denotes element-wise multiplication.

We use a multi-layer LSTM with hidden size $H = 256$ and $L = 4$ layers, followed by dropout with probability 0.25 between layers to mitigate overfitting. After processing the full window of length W, we take the final hidden state $\mathbf{h}_W$ from the top LSTM layer and pass it through a fully connected layer:

$$\hat{\tilde{y}}_{t+1}^{(s,m)} = \mathbf{w}_{\text{out}}^\top \mathbf{h}_W + b_{\text{out}},$$

which is then inverse-scaled to obtain $\hat{y}_{t+1}^{(s,m)}$ in the original units.

The LSTM parameters are learned by minimizing the mean squared error (MSE) over the training set using the Adam optimizer:

$$\mathscr{L}_{\text{MSE}} = \frac{1}{N} \sum_{(s,m)} \sum_{t \in \mathscr{T}_{\text{train}}^{(s,m)}} \left(\tilde{y}_{t+1}^{(s,m)} - \hat{\tilde{y}}_{t+1}^{(s,m)}\right)^2.$$

2) Gated Recurrent Unit (GRU): Gated Recurrent Units (GRUs) [**?**] are a simplified variant of LSTMs with fewer gates and no explicit cell state, leading to a lighter architecture while retaining the ability to model temporal dependencies. For each time step τ, the GRU updates its hidden state $\mathbf{h}_\tau$ according to:

$$\begin{aligned}
\mathbf{z}_\tau &= \sigma\left(\mathbf{W}_z \begin{bmatrix} \mathbf{h}_{\tau-1} \\ x_\tau \end{bmatrix} + \mathbf{b}_z\right), \\
\mathbf{r}_\tau &= \sigma\left(\mathbf{W}_r \begin{bmatrix} \mathbf{h}_{\tau-1} \\ x_\tau \end{bmatrix} + \mathbf{b}_r\right), \\
\tilde{\mathbf{h}}_\tau &= \tanh\left(\mathbf{W}_h \begin{bmatrix} \mathbf{r}_\tau \odot \mathbf{h}_{\tau-1} \\ x_\tau \end{bmatrix} + \mathbf{b}_h\right), \\
\mathbf{h}_\tau &= (1-\mathbf{z}_\tau) \odot \mathbf{h}_{\tau-1} + \mathbf{z}_\tau \odot \tilde{\mathbf{h}}_\tau.
\end{aligned}$$

We use a stacked GRU with hidden size and depth comparable to the LSTM baseline. The final hidden state $\mathbf{h}_W$ is mapped to a scalar forecast via a dense output layer as in the LSTM. Training uses the same loss and optimization procedure.

3) Feedforward Neural Network (FNN): The Feedforward Neural Network (FNN) baseline treats the W-day window as a static feature vector and does not maintain recurrent state. For each (s,m) and time index t, the input is the flattened, scaled vector $\mathbf{x}_t^{(s,m)} \in \mathbb{R}^W$.

We construct an FNN with L_{fc} fully connected layers. The first hidden layer computes:

$$\mathbf{h}^{(1)} = \phi\left(\mathbf{W}^{(1)}\mathbf{x}_t^{(s,m)} + \mathbf{b}^{(1)}\right),$$

where $\phi(\cdot)$ is a non-linear activation function (e.g., ReLU). For $l = 2,\ldots,L_{\text{fc}}$, subsequent hidden layers are defined as:

$$\mathbf{h}^{(l)} = \phi\left(\mathbf{W}^{(l)}\mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}\right).$$

The output layer maps the final hidden representation to the scaled forecast:

$$\hat{\tilde{y}}_{t+1}^{(s,m)} = \mathbf{w}^{(\text{out})\top}\mathbf{h}^{(L_{\text{fc}})} + b^{(\text{out})},$$

which is then inverse-transformed to obtain $\hat{y}_{t+1}^{(s,m)}$. The FNN is trained with the same MSE loss as the recurrent models.

4) Linear Regression: Linear regression serves as a simple, interpretable baseline model that assumes a linear relationship between the next-day cost and its W lagged values. For each (s,m) series, the model is:

$$\hat{y}_{t+1}^{(s,m)} = \beta_0^{(s,m)} + \sum_{k=0}^{W-1} \beta_{k+1}^{(s,m)} y_{t-k}^{(s,m)},$$

where $\beta_0^{(s,m)}$ is the intercept and $\beta_1^{(s,m)},\ldots,\beta_W^{(s,m)}$ are coefficients associated with the lagged costs. The parameters are estimated by minimizing the sum of squared errors over the training set:

$$\min_{\beta^{(s,m)}} \sum_{t \in \mathcal{T}_{\text{train}}^{(s,m)}} \left(y_{t+1}^{(s,m)} - \hat{y}_{t+1}^{(s,m)}\right)^2.$$

While linear regression cannot capture nonlinear or complex temporal patterns, it provides a useful reference point to measure the added value of more expressive models.

5) XGBoost Regressor: Extreme Gradient Boosting (XGBoost) is a tree-based ensemble method that builds a sequence of decision trees to minimize a differentiable loss function with regularization [**?**]. For each example $\mathbf{x}_t^{(s,m)}$, the XGBoost regressor predicts:

$$\hat{y}_{t+1}^{(s,m)} = \sum_{k=1}^{K} f_k\left(\mathbf{x}_t^{(s,m)}\right), \quad f_k \in \mathscr{F},$$

where each f_k is a regression tree and $\mathscr{F}$ is the space of all possible trees with bounded depth. Let N denote the number of training examples for series (s,m). The objective optimized by XGBoost is:

$$\mathscr{L} = \sum_{i=1}^{N} \ell(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k),$$

where $\ell(\cdot,\cdot)$ is the loss function (here, squared error) and $\Omega(f_k)$ is a regularization term penalizing tree complexity, typically of the form:

$$\Omega(f_k) = \gamma T_k + \tfrac{1}{2}\lambda \sum_{j=1}^{T_k} w_{k,j}^2,$$

with T_k the number of leaves in tree k and $w_{k,j}$ the leaf weights. XGBoost can capture nonlinear relationships between lagged costs and the forecast, but it does not maintain an explicit temporal state like recurrent neural networks.

F. Evaluation Metrics

To evaluate the forecasting accuracy of the univariate models developed for each school–meal time series, we employ three standard regression metrics: Mean Squared Error (MSE), Mean Absolute Error (MAE), and the coefficient of determination (R^2). These metrics quantify prediction quality across different dimensions, including average error magnitude, robustness to outliers, and explanatory power.

1) Mean Squared Error (MSE): MSE penalizes large prediction errors more heavily, making it useful for identifying models that occasionally produce large deviations:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2.$$

2) Mean Absolute Error (MAE): MAE measures the typical magnitude of forecast errors and is more interpretable in real-world cost units:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|.$$

3) Coefficient of Determination (R^2): The R^2 metric quantifies how well a model explains variability in daily production costs. Higher values indicate better predictive performance:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}.$$

Model	Multivariate			Univariate		
	MSE	MAE	R^2	MSE	RMSE	R^2
Linear Regression	662.3026	13.7807	0.44	39013.54	197.52	0.75
XGBoost	506.5291	11.7514	0.58	39595.32	198.99	0.74
FNN	677.3214	12.3852	0.43	39241.72	198.10	0.75
LSTM	259.8	14.2605	0.81	18793.44	137.08	0.86
GRU	213.1059	10.3730	0.77	20186.30	142.07	0.85

TABLE III
PERFORMANCE COMPARISON OF MULTIVARIATE AND UNIVARIATE TIME-SERIES FORECASTING MODELS.

Model	MSE	RMSE	R^2
Linear Regression	39013.54	197.52	0.75
XGBoost	39595.32	198.99	0.74
FNN	39241.72	198.10	0.75
LSTM	18793.44	137.08	0.86
GRU	20186.30	142.07	0.85

TABLE IV
UNIVARIATE FORECASTING PERFORMANCE ACROSS ALL SCHOOL–MEAL SERIES.

4) Model Performance: Table III reports the MSE, MAE, RMSE, and R^2 scores for all models evaluated in this study, including both their multivariate and univariate implementations. While multivariate experiments were conducted to assess how models behave when provided with multiple input features, the univariate setting is the primary focus of this work, as forecasting relies solely on the historical production cost sequence for each school–meal series.

Across all models, the univariate LSTM consistently achieved the lowest error and highest R^2, demonstrating superior ability to capture short-term temporal patterns in daily cost trajectories. Consequently, the univariate LSTM is selected as the final forecasting model for all downstream prediction, backtesting, and cost optimization analyses.

G. Evaluation Metrics and Model Comparison

We evaluate all univariate forecasting models using standard regression metrics on the test set for each school–meal time series (s,m). Let $\{y_t^{(s,m)}\}_{t\in T_{\text{test}}^{(s,m)}}$ denote the true daily production costs and $\{\hat{y}_t^{(s,m)}\}_{t\in T_{\text{test}}^{(s,m)}}$ the corresponding model predictions. The mean squared error (MSE) and root mean squared error (RMSE) are defined as:

$$\text{MSE}^{(s,m)} = \frac{1}{|T_{\text{test}}^{(s,m)}|}\sum_{t\in T_{\text{test}}^{(s,m)}}\left(y_t^{(s,m)} - \hat{y}_t^{(s,m)}\right)^2, \tag{1}$$

$$\text{RMSE}^{(s,m)} = \sqrt{\text{MSE}^{(s,m)}}. \tag{2}$$

The mean absolute error (MAE) is:

$$\text{MAE}^{(s,m)} = \frac{1}{|T_{\text{test}}^{(s,m)}|}\sum_{t\in T_{\text{test}}^{(s,m)}}\left|y_t^{(s,m)} - \hat{y}_t^{(s,m)}\right|. \tag{3}$$

The coefficient of determination (R^2) is:

$$R^2_{(s,m)} = 1 - \frac{\sum_{t\in T_{\text{test}}^{(s,m)}}\left(y_t^{(s,m)} - \hat{y}_t^{(s,m)}\right)^2}{\sum_{t\in T_{\text{test}}^{(s,m)}}\left(y_t^{(s,m)} - \bar{y}^{(s,m)}\right)^2}, \tag{4}$$

where $\bar{y}^{(s,m)}$ is the mean production cost in the test set.

To compare performance across univariate models, we compute the average metrics over all school–meal series. Table IV summarizes the results. The univariate LSTM consistently achieves the lowest MSE and RMSE and the highest R^2, with representative R^2 values around 0.86. GRU, FNN, XGBoost, and Linear Regression exhibit lower accuracy, demonstrating that the LSTM's gated memory mechanism more effectively captures short-term temporal dependencies in production costs. Therefore, the univariate LSTM is selected as the final forecasting model for all downstream prediction and backtesting procedures.

H. Train–Validation–Test Split

For each school–meal time series (s,m), we partition the data chronologically to preserve the temporal ordering and avoid information leakage. Let $T_{s,m}$ denote the total number of observations in the series. We use the earliest 60% of the observations for training, the next 20% for validation, and the final 20% for testing.

The training set is used for model fitting, the validation set is used for hyperparameter tuning (e.g., learning rate, hidden dimension, number of trees), and the test set is held out for final evaluation. Importantly, no future values of the time series are used during training or validation, ensuring that all forecasting is performed strictly in a forward-looking manner.

I. Backtesting Procedure

To evaluate forecasting performance on unseen data, we employ a walk-forward (rolling-origin) backtesting strategy. After training each model on the training set and selecting hyperparameters using the validation set, we generate one-step-ahead forecasts sequentially across the test window.

For each test index $t \in T_{\text{test}}^{(s,m)}$, the model receives the previous W observations $\{y_{t-W+1}^{(s,m)},\ldots,y_t^{(s,m)}\}$ and predicts the next-day value $\hat{y}_{t+1}^{(s,m)}$. This procedure mirrors operational deployment, where only past data are available at prediction time.

The resulting forecast sequence $\{\hat{y}_t^{(s,m)}\}$ is then compared against the true values using the metrics described in Section 5.7.

V. RESULTS

Our results evaluate the performance of five forecasting models—Linear Regression (LR), XGBoost, Feed-Forward Neural Network (FNN), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU) across both multivariate and univariate settings. All models were trained using 80% of the available dataset and evaluated on the remaining 20%. The evaluation metrics used include Mean Squared Error (MSE), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and the coefficient of determination (R^2), summarized previously in Table III.

Overall, the multivariate setting exposes performance differences under richer feature representations, whereas the univariate setting evaluates the ability of each model to capture

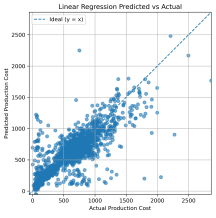

Fig. 4. * Linear Regression

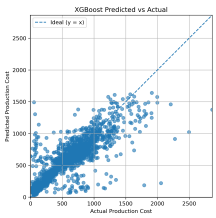

Fig. 5. * XGBoost

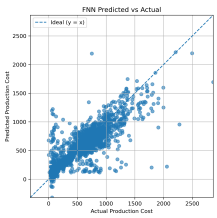

Fig. 6. * FNN

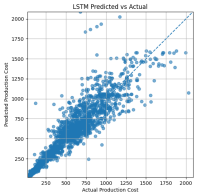

Fig. 7. * LSTM

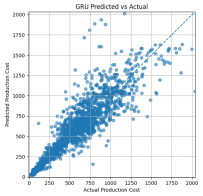

Fig. 8. * GRU

Fig. 9. Univariate Models: Predicted vs. Actual Comparison

temporal structure from the production cost series alone. Across both formulations, the LSTM model exhibits the strongest predictive accuracy and stability, and therefore we adopt it as the primary forecasting model for downstream cost prediction.

A. Multivariate Results

The multivariate analysis evaluates how each model performs when using the full set of engineered features rather than only past production cost values. This setting allows the models to learn richer temporal and contextual patterns linked to meal production cost.

Across all models, the results show clear differences in how effectively each approach captures relationships among multiple input variables. LSTM and GRU demonstrate the strongest ability to model nonlinear temporal dependencies, leading to more accurate cost forecasts. XGBoost and the Feedforward Neural Network (FNN) perform moderately well, benefiting from the additional features but still exhibiting some inconsistency in capturing complex temporal structure. Linear Regression shows the highest prediction error, indicating that linear relationships alone are insufficient to model the multivariate dynamics present in the data.

Overall, incorporating multivariate inputs enhances performance for models capable of capturing nonlinear and sequential patterns, with recurrent neural networks (LSTM and GRU) providing the most reliable predictions.

B. Univariate Results

Figures 9 show the predicted vs. actual cost for the univariate setting, where only past production cost values were used as input. Across all models, the univariate LSTM provides the tightest clustering around the ideal line, indicating the strongest temporal modeling ability.

The univariate LSTM clearly outperforms the other models, with predictions most closely aligned to the ideal reference line. This strong performance validates its selection as the primary forecasting architecture used for the final school meal production cost forecasting experiments presented in Section 9.

C. Forecasting Example Using the Best Model (LSTM, Univariate)

To validate model performance on real forecasting tasks, we generated forward predictions using the univariate LSTM model. The data were split into training and testing segments, and multi-step forecasts were produced beyond the test horizon.

Figure 10 shows an example forecast for *Aldrin Elementary (Lunch)*. The LSTM successfully tracked overall test-set dynamics and produced realistic short-horizon forecasts. This demonstrates its capacity to support operational planning for school meal cost estimation.

In addition to visual inspection, we evaluated the model quantitatively using a 10-step ahead backtesting procedure. The univariate LSTM achieved a mean squared error (MSE) of 14,611.93, a root mean squared error (RMSE) of 120.88, and an R^2 score of approximately 0.89. These results confirm that the model captures underlying cost dynamics effectively and provides reliable short-term forecasting accuracy.

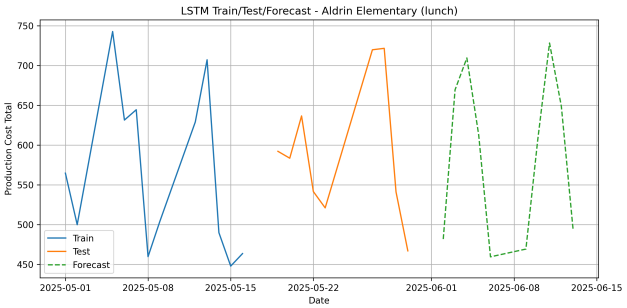

Fig. 10. Univariate LSTM Train/Test/Forecast Example for Aldrin Elementary (Lunch).

Overall, across all schools and meal categories, the univariate LSTM demonstrated the most robust predictive ability, leading us to adopt it as the final model for all downstream forecasting and cost-projection analyses.

Discussion

The primary objective of this study was to evaluate the effectiveness of various machine learning and deep learning models for forecasting school-level production costs within the National School Lunch Program (NSLP). The modeling strategy emphasized univariate time-series forecasting, where only the historical values of production costs were used as predictors. Multivariate experiments were conducted solely for exploratory comparison; however, they did not produce significant improvements and were therefore not prioritized in the final forecasting framework.

Across all models Linear Regression, XGBoost, Feedforward Neural Network (FNN), GRU, and LSTM-the univariate LSTM consistently outperformed the alternatives in terms of prediction accuracy and stability. Its ability to capture long-term

dependencies and nonlinear temporal patterns was particularly advantageous in handling the noisy and irregular nature of school production cost data. The superior performance of the LSTM aligns with findings in past literature demonstrating its robustness for sequential forecasting tasks [3], [8], [9].

To validate forecasting behavior in a realistic setting, a portion of the dataset was split into training and testing subsets. This allowed evaluation of the model's ability to generalize to unseen data before performing forward forecasting. Figure 10 illustrates an example from Aldrin Elementary (Lunch), showing training data, held-out test data, and multi-step forecasts generated by the best-performing univariate LSTM model. This visual confirmation further supports the quantitative metrics by demonstrating the LSTM's ability to follow the underlying trend and variability present in real operational conditions.

Overall, the results strongly indicate that univariate LSTM forecasting is a reliable and effective approach for predicting short-term school-level production costs. This demonstrates the potential for machine learning–based planning tools to support cost estimations, reduce waste, and improve resource allocation in public school meal programs.

VI. Conclusion

This research demonstrates that deep learning methods particularly univariate LSTM networks provide substantial improvements over traditional machine learning models for forecasting production costs in school meal programs. While simpler models such as Linear Regression and XGBoost can capture general cost trends, they lack the sequential learning capability required to model temporal dependencies inherent in financial and operational data. The LSTM model proved most effective at learning temporal patterns from limited and irregularly spaced historical data.

The forecasting framework developed in this study can support operational decision-making by providing short-term cost predictions at the school level. These insights can help school administrators minimize waste, optimize procurement, and better align resources with daily meal production needs. As food waste and cost efficiency remain critical concerns in the NSLP [10], [12], forecasting tools such as the one developed here can contribute meaningfully to data-driven school nutrition management.

A. Limitations

Several limitations should be noted when interpreting the results of this study:

- **Limited historical data:** Many schools provided only short time spans of cost data, restricting the model's ability to learn seasonal or long-term patterns.
- **Lack of exogenous variables:** The study primarily focused on univariate modeling. External factors such as enrollment changes, menu variations, or supplier price fluctuations were not incorporated.
- **Short-term forecasting horizon:** The models were evaluated only for short forward windows due to data constraints. Long-term forecasting accuracy remains untested.
- **Operational variability:** Factors such as special events, shortages, or staffing patterns common in real cafeteria environments were not explicitly modeled.

B. Future Scope

Future research can extend the current study in several impactful directions:

- **Multivariate forecasting:** Incorporating exogenous features such as student attendance, menu type, ingredient prices, or seasonal effects may improve predictive accuracy.
- **Long-term forecasting:** Exploring multi-horizon forecasting models such as Seq2Seq, Temporal Fusion Transformers (TFT), or N-BEATS.
- **Cross-school transfer learning:** Leveraging similarities between schools to improve performance where historical data is scarce.
- **Real-time deployment:** Integrating the forecasting system into school nutrition management software for daily operational decision support.
- **Optimization modules:** Combining forecasting with prescriptive analytics to produce procurement recommendations and waste-reduction strategies.

References

[1] Brownlee, J. (2020). *Deep Learning for Time Series Forecasting: Predict the Future with MLPs, CNNs and LSTMs in Python*. Machine Learning Mastery.

[2] Chollet, F. (2018). *Deep Learning with Python*. Manning Publications.

[3] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.

[4] Yu, H. F., Huang, F. L., & Lin, C. J. (2011). Dual coordinate descent methods for logistic regression and maximum entropy models. *Machine Learning*, 85(1–2), 41–75.

[5] Scikit-learn developers. (2024). *Scikit-learn: Machine Learning in Python*. Retrieved from https://scikit-learn.org/

[6] PyTorch contributors. (2024). *PyTorch Documentation*. Retrieved from https://pytorch.org/

[7] Kaggle. (2024). Time Series Forecasting Tutorials. Retrieved from https://www.kaggle.com/

[8] Kong, W., Dong, Z. Y., Jia, Y., Hill, D. J., Xu, Y., & Zhang, Y. (2019). Short-term residential load forecasting based on LSTM recurrent neural network. *IEEE Transactions on Smart Grid*, 10(1), 841–851.

[9] Lai, G., Chang, W. C., Yang, Y., & Liu, H. (2018). Modeling long- and short-term temporal patterns with deep neural networks. In *SIGIR* (pp. 95–104).

[10] Mulhollem, J. (2021). U.S. school cafeterias waste more food than those in other developed countries. *Penn State University News*. Retrieved from https://www.psu.edu/news/

[11] Ralston, K., Treen, K., Coleman-Jensen, A., & Guthrie, J. (2017). *National School Lunch Program: Trends and Factors Affecting Student Participation*. USDA Economic Research Service.

[12] Smith, S. L., & Cunningham-Sabo, L. (2014). Food choice, plate waste, and nutrient intake of elementary- and middle-school students in the U.S. National School Lunch Program. *Public Health Nutrition*, 17(6), 1255–1263.