Forecasting Food Loss

By Bella Chang, Fiona Chang, Wilson Fung, Sumin Park, Tess U-Vongcharoen, Sandya Wijaya

1

Introduction

1.3 billion tons of food wasted annually worldwide

Economic loss of approximately

$940 billion

due to *food wastage*

Food waste contribute to

8%

of ***global greenhouse gas emissions***

Research Question

What machine learning (ML) techniques can effectively forecast food loss?

Objectives:

- To utilize ML techniques, learned in class and beyond, to characterize national food loss.
- To compare and evaluate the effectiveness of predictive models in forecasting food loss.

Scope:

- Address food loss **within USA** by training on Food and Agriculture Organization's (FAO) dataset.
- Concentrate on **fruit category** among our categorized datasets of roots, vegetables, fruits, and animal products.

Stakeholders + Benefits

Stakeholders:

- Government agencies and policymakers.
- Food supply chain entities and NGOs/NPOs working to address food security.

Benefits:

- **Enhanced forecasting ability** to anticipate and manage food shortages or surpluses.
- Data-driven insights for **strategic planning** and **resource allocation** in food supply chains.
- **Contribution to global efforts** in reducing food loss and achieving food security.

2

Exploration & Dataset

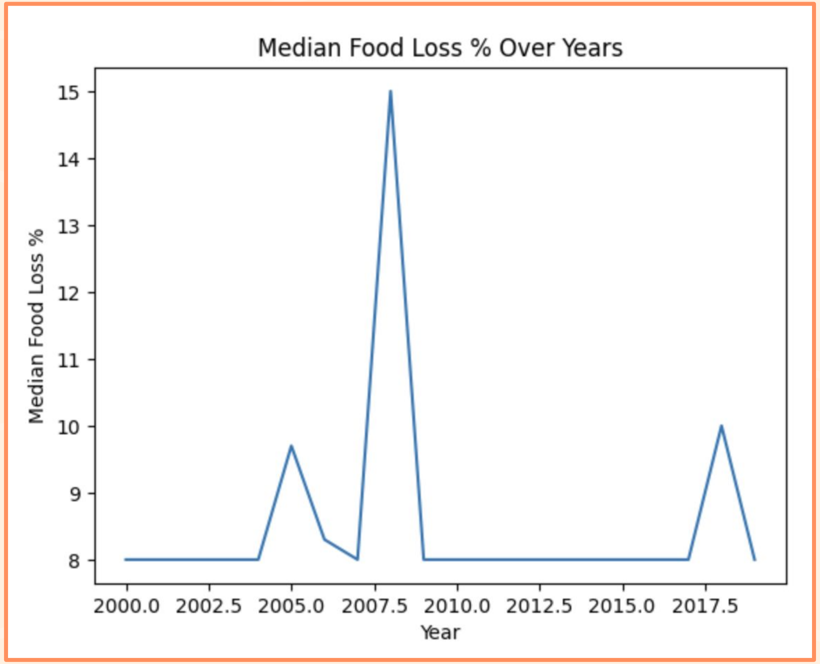
Food & Agricultural Organization of the United Nations (FAO)

Food Loss & Waste Database

country object	region float64	cpc_code float64	commodity object	year int64	loss_percentage fl...
United State... 100%	NaN - NaN	231.0 - 23210.04	Cabbages 7.7% Lettuce and ... 4.8% 69 others 87.6%	2000 - 2019	0.009195652 - 64.0
United States of ...	nan	1211	Asparagus	2019	9
United States of ...	nan	1212	Cabbages	2019	7
United States of ...	nan	1212	Cabbages	2019	8
United States of ...	nan	1212	Cabbages	2019	12
United States of ...	nan	1212	Cabbages	2019	12
United States of ...	nan	1212	Cabbages	2019	12
United States of ...	nan	1213	Cauliflowers and ...	2019	8
United States of ...	nan	1213	Cauliflowers and ...	2019	8
United States of ...	nan	1214	Lettuce and chic...	2019	7

Loss_Percentage is calculated by FAO's Food Loss Metric, which calculates loss from harvest up to but excluding retail stage

Median Loss % Of Food Over Years



Chose **median** food loss % for visualizations and modeling to reduce large influence of outliers

Observations

- Spike in 2007-2009 because of the Great Recession
- High unemployment rate
  → Less Spending of Food
  → Higher Food Loss

Takeaways

- To investigate economic factors in relation to food loss

Feature Exploration

Precipitation	Fertilizer
Crop Yield	GNI Per Capita (US $)
GDP	Temperature (Farenheit)
CO2 Emissions	Year

Data Preprocessing

- **Narrowed Time Scope**: 2000 - 2019
  - Training Data: 2000 - 2015
  - Test Data: 2016 -2019

- **Categorized** FAO's Food Loss Dataset into different categories of food: fruit, vegetables, grains, and meat products
  - Team Decision: Narrow data to fruits for LSTM and ARIMA model

- Data Preprocessing for Models
  - Convert columns into **appropriate data types** for modeling
  - **Scale data** from 0 to 1 using MinMaxScaler
  - **Removing** NaN values
  - **Selecting most important features** by conducting correlation tests and using domain knowledge

3

Methods

Models: FFNNs, LSTM, ARIMAX

1

Feed Forward Neural Network (FFNN):
Baseline, no consideration of time series dependencies

2

Long Short Term Memory (LSTMs):
Recurrent neural network (RNN) we learned in class, incorporates time dependency

3

ARIMAX:
A new model we found through own research, best for time series forecasting

FFNNs

Feed Forward Neural Network

*We decided to utilize **feed forward neural network** because …*

- Can be applied for wide range of tasks including regression
- Capable of learning complex relationships with hidden layers
- Introduce non-linearity to the data
- Simplest model that uses benefits of backpropagation to minimize error

Way it works:

- Takes in an output layer with X features, then runs that through a set of weights for each hidden layer before it outputs 1 number
- Each hidden layer:
  - Can capture hidden hierarchical relationships
  - Can learn which features are more important

Feed Forward Neural Network

- **9** input features, **2** hidden layers with nonlinearity introduced.
- With **69% accuracy**, predicted within **5% of the true food loss percentage.**

```
# Define the architecture of the neural network
model = models.Sequential()

# Add input layer
model.add(layers.Dense(units=64, activation='relu', input_shape=(X_train.shape[1],)))

# Add hidden layers
model.add(layers.Dense(units=128, activation='relu'))
model.add(layers.Dense(units=128, activation='relu'))

# Add output layer with one node to output the percentage loss
model.add(layers.Dense(units=1, activation='linear'))

#rmse loss function
def rmse(y_actual, y_pred):
    return K.sqrt(K.mean(K.square(y_pred - y_actual)))

# Compile the model using an appropriate loss function
model.compile(optimizer='adam', loss='mean_squared_error', metrics=[rmse])

model.summary()

model.fit(X_train, y_train, epochs=20, batch_size=32, validation_split=0.2)
```

Feed Forward Neural Network

Error Metric: MAPE (Mean Absolute Percentage Error)

- MAPE= 0.657 means that on average, predictions are off by about 65.7%.
- In comparison, the RMSE ranged from [8.2, 8.3]

***Disadvantages** of Feed Forward Neural Network*:

- Lack of temporal understanding, meaning the model did not take into account the time series data
  - Limited memory
  - Sensitivity to noise
  - Overfitting when training data is limited
  - Poor at identifying sequential relationships
- Gradient Vanishing unless introducing nonlinearity with functions like ReLU
  - ReLU computationally expensive

LSTM

LSTM Model - Intro

We decided to utilize Long Short-Term Memory (LSTM network) because ...

- Recurrent neural network are designed to handle sequence dependence, and the Long Short-Term Memory (LSTM network) is a type of RNN used in deep learning because very large architectures can be successfully trained.

Way it works:

- LSTM's gates **dynamically learn dependencies across sequences** = allowing it to capture complex temporal patterns and long-range dependencies more effectively.
- LSTM’s memory cells **preserves and process information over varying input sequences** = can handle non-stationary data (data that exhibit trends and thus eligible for time series prediction)
- LSTM gates **control the flow of information** = allowing it to capture and update information in the memory cells based on the input and past memory content.

LSTM Model - Dataset Preparation

- Choose 1 year as the lookback period ie. 2000 to predict 2001 and so on

Target:

- year', 'lost_crop_yield', 'scaled_values' (normalized lost_crop_yield)
    - We used MinMaxScaler to transform our data to values ranging from 0 to 1
- For each year, found the lost crop yield (loss % x food yield) → grouped by year, using median as the aggregate → 1 observation = median crop yield in a single year

Features:

- Indicators to show food growth AND food access
- climate factors, land use, agricultural details, physical factors, socioeconomic factors *

* Full list: CO_2 emissions, fertilizers (N, P_2O_5, K_2O), GDP, Crop land use (harvested, Double cropped, Cropland harvested, Crop failure, Cultivated summer fallow, Total), Precipitation, Rail lines density, Rural vs Urban population, Temperature

LSTM Model - Creation

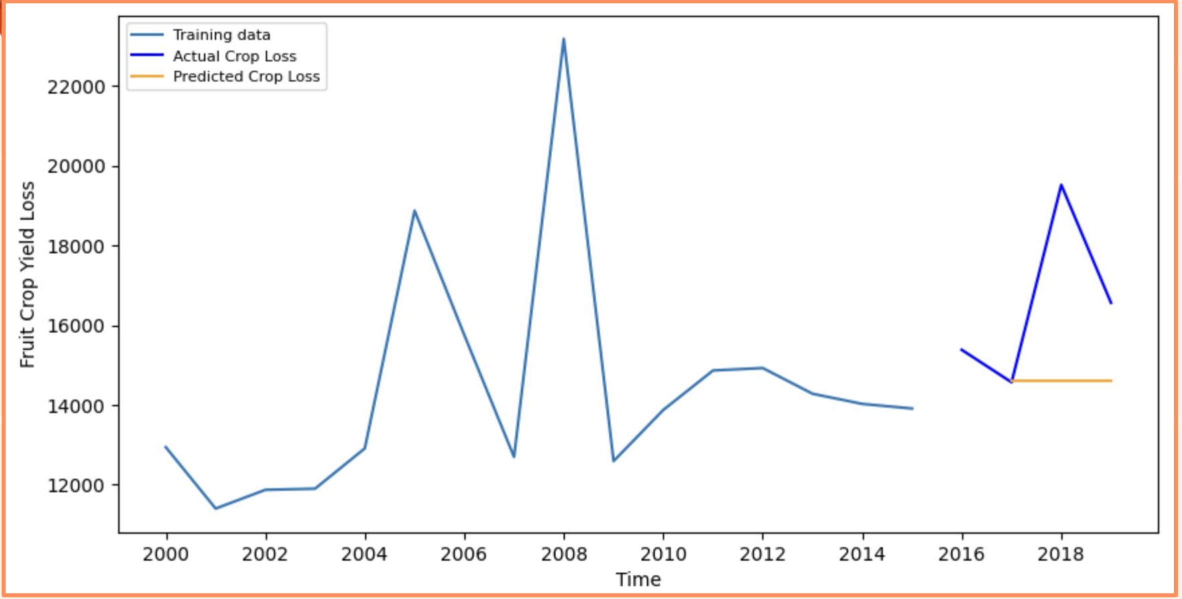
- The model starts with an input layer accommodating sequences of `x_train`; 1 x 18 as there are 18 features.
- **Four LSTM layers** are stacked, each with 50 memory units.
    - All except the last one have **`return_sequences=True`** for information propagation across sequential steps.
    - A **dropout layer with a rate of 0.25** follows every layer to prevent overfitting by randomly deactivating a fraction of neurons during training.
- The final layer is a **Dense layer with a single unit**, applying a **linear activation function**, suitable for regression
- The model is compiled using the **Adam optimizer** and **mean squared error** as the loss function.
- Training the model involves fitting it to `x_train` and `y_train` data for **50 epochs with a batch size of 32**.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 1, 50)             13800

 dropout (Dropout)           (None, 1, 50)             0

 lstm_1 (LSTM)               (None, 1, 50)             20200

 dropout_1 (Dropout)         (None, 1, 50)             0

 lstm_2 (LSTM)               (None, 50)                20200

 dropout_2 (Dropout)         (None, 50)                0

 dense (Dense)               (None, 1)                 51

=================================================================
Total params: 54,251
Trainable params: 54,251
Non-trainable params: 0
_________________________________________________________________
```

```
model.compile(optimizer='adam', loss='mean_squared_error', metrics=['accuracy'])
model.fit(x_train, y_train, epochs = 50, batch_size=32)
```

```
1/1 [==============================] - 0s 64ms/step - loss: 0.0672 - accuracy: 0.0667
Epoch 36/50
1/1 [==============================] - 0s 8ms/step - loss: 0.0732 - accuracy: 0.0667
```

LSTM Model - Results



- **MSE:** 9333441.7485343
- **MAE:** 2302.3615364
- **RMSE:** 3055.068206
- **MAPE:** 0.124140

LSTM Model - Conclusions

Error Metric: MAPE

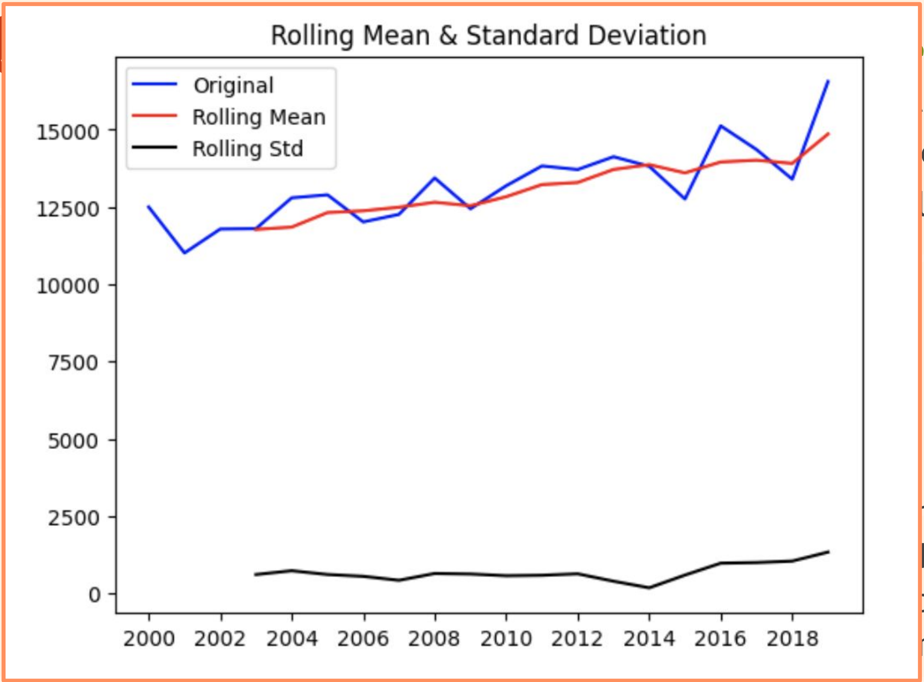
- Since my target is normalized (between 0-1 range), MAPE can understand the average percentage error in predictions compared to the actual normalized lost crop yield for each year.
- MAPE= 0.124 means that on average, predictions are off by about 12.4%.

Disadvantages of LSTM Model:

- **Prone to Overfitting:** LSTMs can overfit on small datasets, especially when the model complexity is high. Regularization techniques like dropout or early stopping are often used to mitigate overfitting, but as seen by our high MSE, they might not always solve this issue.
- **Difficulty Capturing Long-Term Dependencies:** LSTMs are designed to capture long-term dependencies. Since our the food loss prediction task involves intricate, non-linear, & distant relationships between variables, the model struggles to effectively utilize these dependencies.

ARIMAX

ARIMAX Model




```
ADF Statistic: -0.4524523450229775
p-value: 0.9009659245433874
Critical Values:
        1%: -4.223238279489106
        5%: -3.189368925619835
        10%: -2.729839421487603
```

Indicators time series is NOT stationary:

- Rolling mean and rolling STD are still increasing over time, not flat
- P-value of ADF is well above cutoff of 0.05

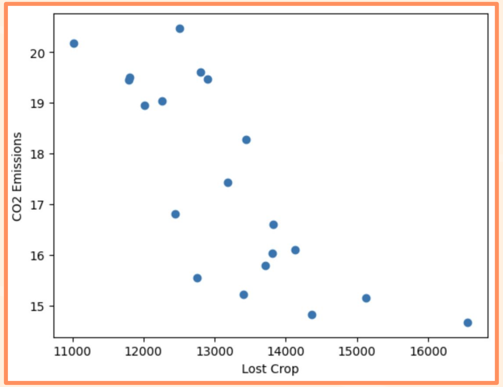
ies to make it stationary
between the current observation and a
plied to lagged observations
related variables to the target goal to tune
its predictions

ARIMAX Model: Feature Selection

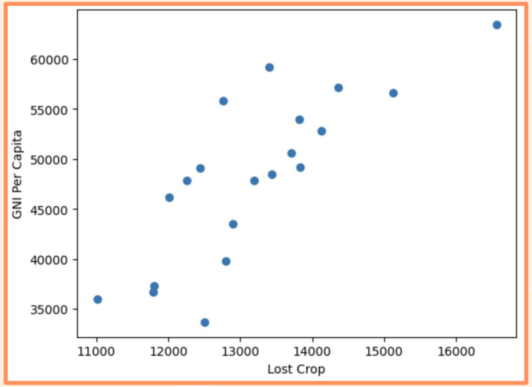
	lost_crop_yield floa...	CO2 Emissions flo...	Fertilizer float64	GDP float64	GNI Per Capita (U...	Annual % Growth fl...	Avg. Precipitation ...
lost...	1	-0.7842748712	0.1512800913	0.7896891773	0.7971038	-0.05964693501	0.4531946447
CO...	-0.7842748712	1	-0.3084715889	-0.87071922	-0.9399826985	0.3194883676	-0.5572347627
Fer...	0.1512800913	-0.3084715889	1	0.2284261899	0.201660821	0.1115083293	0.08059729578
GDP	0.7896891773	-0.87071922	0.2284261899	1	0.9686063634	-0.05385433483	0.664144509
GNI...	0.7971038	-0.9399826985	0.201660821	0.9686063634	1	-0.1748007783	0.6351348402
An...	-0.05964693501	0.3194883676	0.1115083293	-0.05385433483	-0.1748007783	1	-0.02269290026
Avg...	0.4531946447	-0.5572347627	0.08059729578	0.664144509	0.6351348402	-0.02269290026	1

ARIMAX Model: Feature Selection

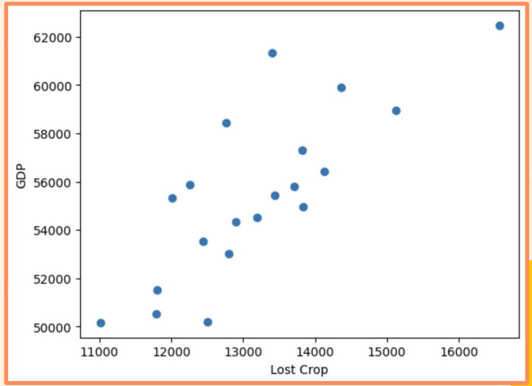
CO2 Emissions



GNI Per Capita



GDP



ARIMAX Model

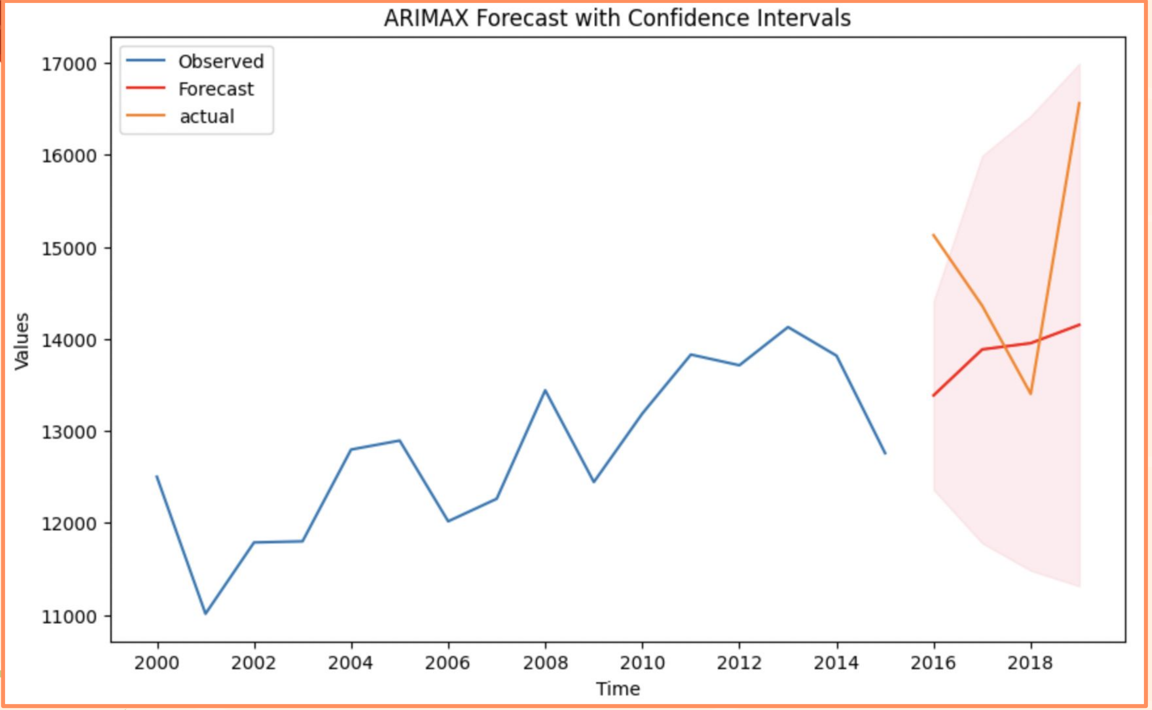
1. Split into train/test
2. Fit ARIMA model on train with exog variables of high correlation
3. ARIMA will iteratively train to a given precision
4. Can validate using same error metrics as NNs

$$d_t = c + \sum_{n=1}^{p} \alpha_n d_{t-n} + \sum_{n=1}^{q} \theta_n \epsilon_{t-n} + \sum_{n=1}^{r} \beta_n x_{n_t} + \sum_{n=1}^{P} \phi_n d_{t-sn} + \sum_{n=1}^{Q} \eta_n \epsilon_{t-sn} + \epsilon_t$$

SARIMAX formula

```
Machine precision = 2.220D-16
 N =            8     M =           10

At X0         0 variables are exactly at the bounds

At iterate    0    f=  5.90094D+00    |proj g|=  7.03763D-01

At iterate    5    f=  5.87448D+00    |proj g|=  1.03338D-01

At iterate   10    f=  5.87252D+00    |proj g|=  8.19434D-02

At iterate   15    f=  5.84763D+00    |proj g|=  2.71012D-01

At iterate   20    f=  5.82192D+00    |proj g|=  6.81501D-02

At iterate   25    f=  5.82038D+00    |proj g|=  4.40665D-03

           * * *
```

ARIMAX Model: Results



- **MSE:** 2339443.740301969
- **MAE:** 1293.4475868633795
- **RMSE:** 1529.5240241009517
- **MAPE:** 0.07956202482648135

ARIMAX Model

Error Metric: MAPE

- Easy to tell overall % of inaccuracy within predictions, esp when compared with LSTM (also being gauged using MAPE)
- MAPE= 0.079 means that on average, predictions are off by about 7.9%

***Disadvantages** of ARIMAX Model*:

- Must find exogenous variables through given predicted forecast interval
- One model must constantly be refit and regenerated an order => inherently dynamic
- Limited forecast horizon
- Sensitive to outliers

4

Results and Conclusions

Model	Advantages	Disadvantages	Error Metrics
[Baseline] Feed Forward Neural Network	• Applicable for wide range of tasks • Can learn complex relationships with hidden layers • Handles non-linearity to the data	• Lack of memory • Fixed input size • Sensitive to noise • Black box nature • Risk of Overfitting	MSE: 115.48021 MAE: 6.31144 RMSE: 10.74617 MAPE: 0.65704 Accuracy: 69%
LSTM Model	• Robust to noisy data • Auto Feature Extraction • Parallelization	• Complex, needs more resources • Risk of Overfitting • Memory Efficiency Risk	MSE: 9333441.74854 MAE: 2302.36154 RMSE: 3055.06821 MAPE: 0.12414
ARIMAX Model	• Takes time into account • Interpretable • Improves with more data • Optimizes own iterative process	• Assumes Linearity • Sensitive to Outliers • Limited Forecast Horizon	MSE: 2339443.74030 MAE: 1293.44759 RMSE: 1529.52402 MAPE: 0.07956

Consolidated Findings

Feed forward neural network was a good start, but **did not capture the complexity** of the underlying causes & time well

LSTMs captured more of these needed time dependencies, but the **model's high error** proved it is overall **better for long term predictions**

ARIMAX **prioritizes trends of time** which most directly affect models like these, adding in exogenous variables as features to further tune general trend and resulting in **lowest error**

Overall, trying different models showed the effect of 1) model complexity 2) targeting exact features / areas. This affected ability for a model to be ***consistently accurate long term***.

Implications and Results

- Knowledge from forecasting food loss encourages:
  - Suppliers to **re-evaluate supply chains**
  - Farmers to **plan food production ahead of time**

- **Increases awareness** of food loss and encourages suppliers to take action

- Action will result in **less food loss** and **increased financial savings**

Recommendations

Reproducible / Applicability:

- Continue updating the model each year with new supplemental data
- NGOs: Ready to be used on a **smaller-scale** to provide insights
- Government Agencies / Policy Makers: Not ready to be used
  - Should increase accuracy more for **large-scale decisions**

Future Improvements:

- Work to decrease MAPE to 0.02 -> on average, predictions are off by 2%
- Work to develop a classifier to help remove outliers in pre-processing

5

Questions?

Sources

- ARIMA vs LSTM on NASDAQ stock exchange data: https://www.sciencedirect.com/science/article/pii/S1877050922013382#:~:text=The%20longer%20the%20data%20window,1.8%20times%20better%20than%20LSTM

- FAO: SDG Indicators Portal https://www.fao.org/sustainable-development-goals/indicators/1231/en/

- Time Series Prediction: Neural Networks https://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras/

- Understanding Climate Impacts on Food Supply https://climatechange.chicago.gov/climate-impacts/climate-impacts-agriculture-and-food-supply