Performance of ARIMA and Convolutional Neural Network Models for Sales Forecasting

Sarah Waterman
Department of Resource Analysis, Saint Mary's University of Minnesota, Winona, MN 55987

Keywords: Sales, Forecasting, Projections, Machine Learning, Statistics, Python

Abstract

Sales forecasting provides insights that allow businesses to plan for the upcoming months and years by allocating funds to the proper areas of the company. This project compared a traditional statistical method of forecasting (ARIMA) to a more complex machine learning method (CNN) when forecasting sales at the Customer and SKU level for an unnamed company. Analysis was conducted using Python and various machine learning and statistical Python packages. It was found that ARIMA almost always results in better RMSE, MAE, and MAPE scores for the data used in this project, and it may be best suited for everyday use within the company since it is less complex than CNN.

Introduction

Sales forecasting is the base of any business plan (Fabianová, Kačmáry, Molnar, and Michalik, 2016). Visibility to future sales allows businesses to plan for reducing costs, marketing, securing trustworthy supply chains, transportation needs, and even labor requirements (Mentzer and Bienstock, 1998, as cited in Chu and Zhang, 2003; Ramos, Santos, and Rebelo, 2015). This project will compare and evaluate two forecasting models in several scenarios for an anonymous company, one using machine learning and one using statistics.

Background

Although sales forecasting is exceptionally influential and crucial for businesses, choosing or developing models which output accurate forecasts remains problematic (Fabianová *et al.*, 2016). Fabianová *et al*. (2016) explain that the accuracy of any forecast heavily depends on the method used, the quality of the input data, the time period for which one is forecasting, and a myriad of other factors, most of which are affected by market uncertainties. Some of these factors are out of the control of a given business, according to Fabianová *et al*. (2016), including the sell price of competing products, the cost of inputs such as materials and labor, and inflation. These factors, along with strong trend and seasonal variations present in most sales time series make it challenging to develop forecasting models that are adequately effective (Ramos *et al*., 2015). According to Lalou, Ponis, and Efthymiou (2020), the complexity of sales forecasting has further increased in the last several years due to the widespread use of e-commerce and the COVID-19 pandemic.

Along with many factors of influence to consider, there are also numerous forecasting methods to choose from. These methods can be broken into two major categories, linear and non-linear (Chu and Zhang, 2003; Caglayan, Satoglu,

Sarah Waterman. 2022. Performance of ARIMA and Convolutional Neural Network Models for Sales Forecasting. Volume 25, Papers in Resource Analysis. 16 pp. Saint Mary's University of Minnesota University Central Services Press. Winona, MN. Retrieved (date). https://gis.smumn.edu/

and Kapukaya, 2020). Linear forecasting usually involves simpler statistical methodology and only considers historical sales data, while non-linear methodology often requires machine learning and considers many factors as inputs (Ma and Fildes, 2021). According to Caglayan *et al*. (2020), each of these schools of methodology can perform differently within different contexts based on which factors of influence are present, so choosing the best method possible for a given context is of utmost importance.

Aside from complications with choosing a model, it is difficult to implement and maintain a model once it is created (Lalou *et al*., 2020). Lalou *et al*. (2020) propose that managers usually put methods into practice that are rather easy to use and that they completely understand, whether or not they are the best methods for a given situation. Usually, Lalou *et al*. (2020) continue, this results in management using linear forecasting methods. If they were to use a non-linear method, it would likely require tools with an easy-to-understand user interface that forecasts sales using just a few inputs from the user (Lalou *et al*., 2020). The licensing and maintenance of these tools can become expensive and labor-intensive, which acts as a barrier to the adoption of more complex, non-linear forecasting methods (Lalou *et al*., 2020).

Project Intent

This project explored two methods to forecast sales for an anonymous company. To protect data privacy in this project, the company will be referred to as Company A. Company A does not currently have a standardized way to forecast sales, so creating a systematic and data-based approach will allow them to create a business plan that includes targeting customers and SKUs with more potential in terms of sales, potentially resulting in increased revenue over time.

The two methods that have been chosen for this project are Autoregressive Integrated Moving Average (ARIMA) and Convolutional Neural Network (CNN). Both methods were conducted at the Customer and SKU levels using Python. The forecasts were pooled, meaning there was one prediction for all Customers and one prediction for all SKUs.

ARIMA

ARIMA is a statistical forecasting method that combines autoregressive (AR) and Moving Average (MA) methodology, where the weight of each component method is customizable (Chu and Zhang, 2003). No matter the weights chosen for AR and MA methods, the output of any ARIMA model expresses future sales as a linear combination of past seasonal and nonseasonal observations of a time series (Chu and Zhang, 2003). “Seasonal observations” refers to the cyclical nature of the series observed throughout different seasons, while “nonseasonal observations” refers to the general trend of the series and other random fluctuations. ARIMA was chosen as one of the forecasting methods for this project, because literature shows that it is generally the most effective statistical forecasting method. If ARIMA performs adequately for Company A, then it could serve as a low-maintenance, low-cost, and time effective method for forecasting sales without requiring more extensive knowledge as machine learning models do.

CNN

There are several Neural Network (NN) forecasting models, all of which are based

on the way the human brain processes information (Caglayan *et al.*, 2020; Chu and Zhang, 2003; Pan and Zhou, 2020). Caglayan *et al.* (2020) cite Efendigil, Önüt, and Kahraman, (2009), explaining that NN models contain neurons which are "connected to their neighbors with varying coefficients which indicate connectivity strength among them." These neurons (or nodes) represent the variables that affect sales outcomes, and they must be chosen by the user (Chu and Zhang, 2003). The NN then uses these neurons and their connections to derive patterns and trends from the data and make predictions for the future (Chu and Zhang, 2003). Many, although not all, of the NN models require an initial selection of nodes but can then work without intervention, automatically determining which nodes are most influential and eventually deriving a forecast (Caglayan *et al*., 2020; Pan and Zhou, 2020). The CNN model was chosen for this project because literature shows that it has stronger usability and does not require extensive knowledge of the data as some other models do, which is ideal for corporate settings (Pan and Zhou, 2020).

Methods

The methods used for this project included an initial analysis of the data, data preparation, organizing the data into various structures to best feed the analyses, and finally an ARIMA analysis and a CNN analysis with each of the created structures.

Data

The data used for these analyses was extracted from the Company A database. One hundred each of Customers and SKUs were provided by Company A, and they were chosen using simple randomization. These Customers and SKUs are not necessarily related, but they could be. Since the Customer and SKU analyses are separate, whether Customers and SKUs are related has no impact on the validity of the analyses. After removing any Customers or SKUs that did not have at least five years of data, the final number of Customers was 95 and the total number of SKUs was 93. According to Luxhoj, Riis, and Stensballe, 1996, as cited in Chu and Zhang (2003), five years of data is sufficient for forecasting purposes, and larger samples are not necessarily helpful. Because of this, the time frame pulled for each Customer and SKU is 1/1/2017 to 4/30/2022, which is just over five years and a total of 64 months. Extracting any more data than this from the database would also require a more rigorous process of checks and balances at Company A. All Customer and SKU identification was masked using Power BI before it was transferred off a company computer. For Customers, customer ID (masked), invoice month, extended sales dollars, average sale price per item, total quantity sold, freight dollars, business days, and number of invoices were obtained. For SKUs, SKU ID, invoice month, extended sales dollars, average sale price per item, total quantity sold, freight dollars, business days, number of invoices, number of customers, and number of branches were obtained.

Data Preparation

The original data used for this project is 2D, but CNN models require 3D data. Thus, to complete a year's worth of predictions, the 2D Customer and SKU datasets had to be divided into yearly subsets to make the dataset 3D. To make it possible to split the data into clean years, the original dataset of 64 months was

decreased to only 60 months (5/1/2017 to 4/30/2022), a total of five years. Although converting the dataset to 3D only needed to be done for the CNN model, the datasets were also reduced to 60 months for the ARIMA model for consistency.

As shown in Figures 1 and 2, the COVID-19 pandemic impacted sales across both time series, particularly April 2020 through August 2020 for Customers and April 2020 through December 2020 for SKUs. Menculini, Marinia, Proietti, Garinei, Bozza, Moretti, and Marconi (2021) removed all 2020 data in their study for this reason. Because of limited data, this project simply replaced the values for April 2020 through August 2020 for the values that were recorded for April 2019 through August 2019. The same process was done for September 2020 through December 2020 for SKUs. This workaround minimized the impact of COVID-19 on model performance as far as possible without decreasing the size of the dataset. The modified datasets are shown in Figures 3 and 4.

Before any other preparation steps were taken, the data was split into a train set and a test set. The train dataset was used to train the model, and the test dataset was used to evaluate how well the model performed. The goal was to produce predictions using the train dataset that were as close as possible to the values in the test dataset. Generally, a split of 70-80% for train and 20-30% for test is used (Temür, Akgün, and Temür, 2019). This project used an 80/20 split, because it nicely divides 60 months of data into even years. For time series predictions the two sets are split chronologically, so the train dataset used in this project was 48 months (May 2017 through April 2021), and the test dataset was 12 months (May 2021 through April 2022). The split between train and test data for Customers and SKUs is shown in Figures 3 and 4.

Figure 1**,** created with the seasonal_decompose function of the *statsmodels* package in Python, shows that the time series for sales dollars across all Customers has significant seasonality and trend. This was confirmed using the Augmented Dikey-Fuller (ADF) test which returns a p-value for significance, as was done in Menculini *et al.* (2021). There is also seasonality and trend for SKU sales (Figure 2), although it is less variable. The ADF test showed that the seasonality and trend for SKUs was not significant enough to render the dataset non-stationary, so there is no need to remove them before forecasting. Although the trend and seasonal components of time series are traditionally removed before forecasting, it is unclear in the literature whether deseasonalization improves or hinders forecast performance (Chu and Zhang, 2003). Thus, both options remain valid. Upon coding the CNN model for Customers, it became clear that leaving the seasonal and trend components within the data would not lead to viable results, so they were both removed from the train dataset for fitting the model and forecasting, then added back to the predictions for comparison to the test data. This was done despite the results of the ADF test because without removing seasonality the model returns forecasts that are very close to the mean without enough variability to be useful for decision making. In that case, the predictions may be more accurate in terms of statistical evaluation, but it was decided that removing seasonality for less statistically accurate but more useful predictions was preferred. As Temür *et al.* (2019) point out, data-compatibility and prediction success should both be considered when choosing a prediction model. The train data for Customer and SKU CNN models

was also normalized and standardized as is standard practice, and the predictions were denormalized and destandardized. No modifications were necessary for the ARIMA model, as these factors were automatically considered in that case.

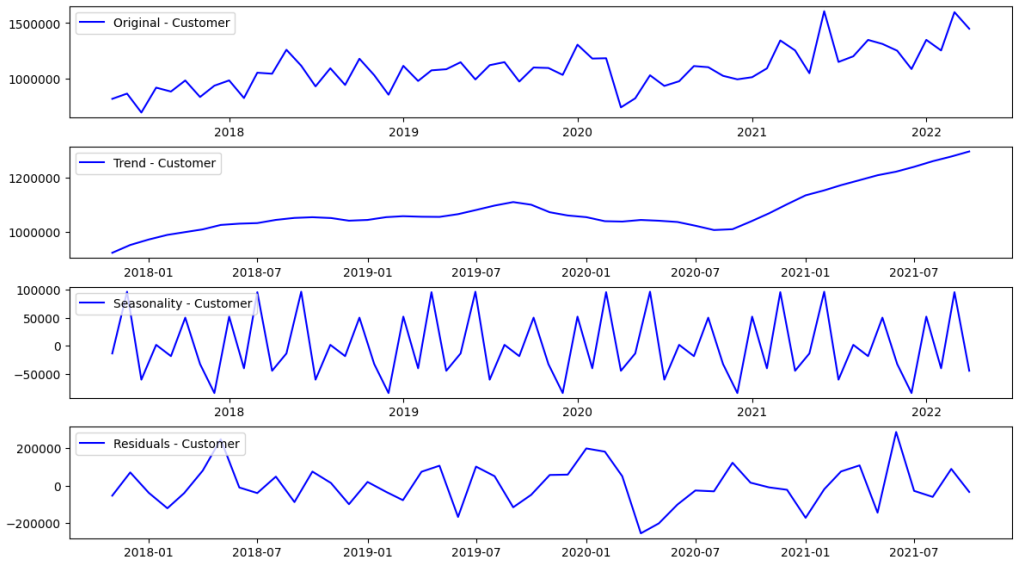

Figure 1. Line charts for all customer original sales time series, trend, seasonality, and residuals generated by the seasonal_decompose function of the *statsmodels* package within Python. The X axis shows the passage of time, and the Y axis shows sales in USD. Significant seasonality and trend are shown, and it is clear that the COVID-19 pandemic had a significant effect on sales during 2020.

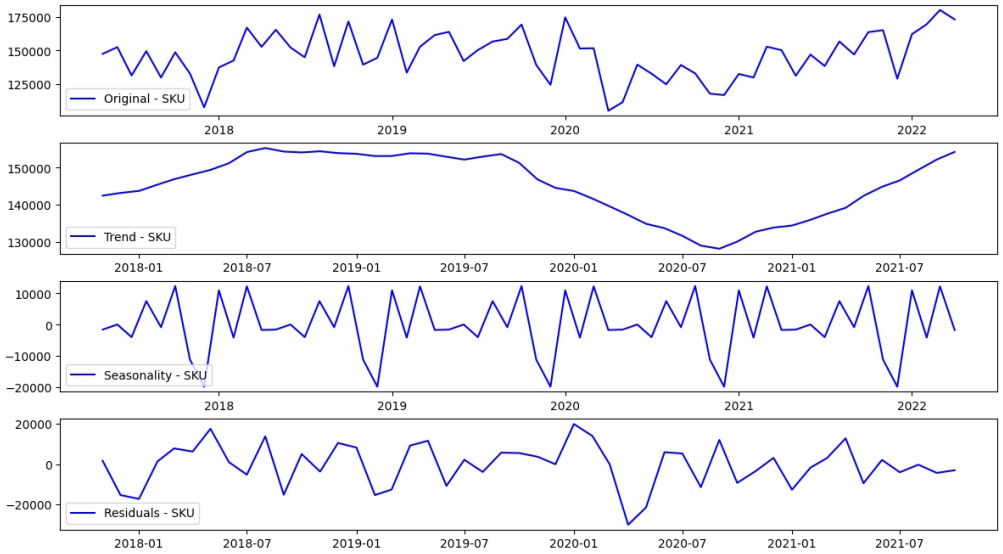

Figure 2. Line charts for all SKU original sales time series, trend, seasonality, and residuals generated by the seasonal_decompose function of the *statsmodels* package within Python. The X axis shows the passage of time, and the Y axis shows sales in USD. Significant seasonality and trend are shown, and it is clear that the COVID-19 pandemic had a significant effect on sales during 2020, so much so that the trend is negative for much of 2020.

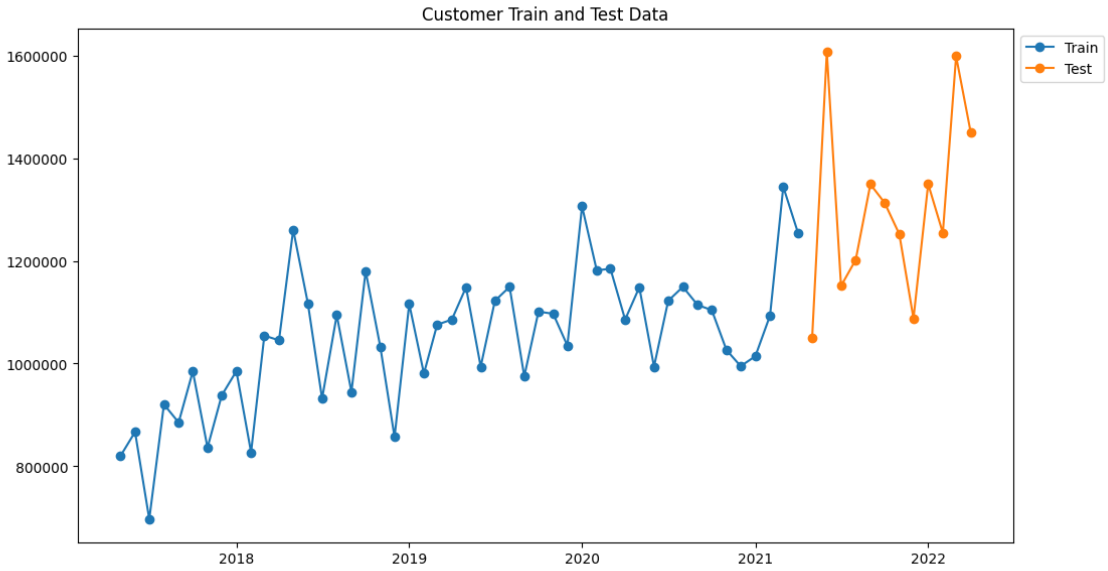

Figure 3. Sales data (USD) for all 95 Customers split into train (first 80% of data) and test (last 20% of data) sets after modifying the values of April 2020 through August 2020 to match the values of April through August of the previous year to accommodate for COVID-19 impacts.

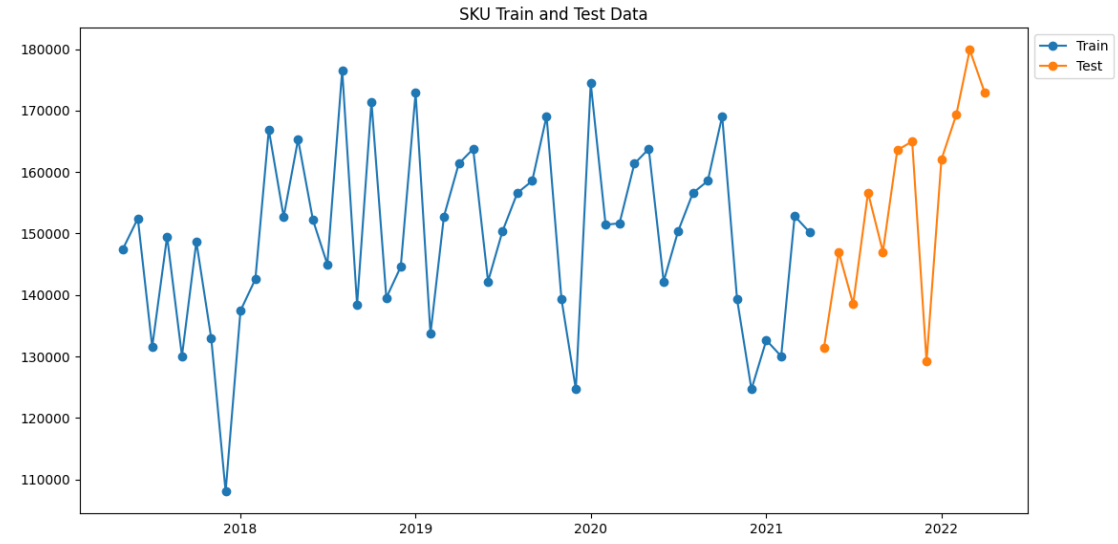

Figure 4. Sales data (USD) for all 93 SKUs (product codes) split into train (first 80% of data) and test (last 20% of data) sets after modifying the values of April 2020 through December 2020 to match the values of April through December of the previous year to accommodate for COVID-19 impacts.

Data Structure

The first data structure simply used the sums and averages for all Customer or all SKU variables per month, but in using this structure the ability to factor in the sales for each individual Customer or SKU is compromised. To solve this issue, the data

was also structured in such a way that each Customer's or SKU's sales became another variable. In addition to these two structures, a final structure combines them to include all individual sales and the other variables. The first structure described was titled Structure 1 (Table 1), the second Structure 2 (Table 2), and the third Structure 3 (Tables 1 and 2 combined). All structures were created using Excel Power Query.

ARIMA Analysis

Because it was determined that the dataset has a significant seasonal component and multiple variables are being used to conduct the forecast, the ARIMA model became a Seasonal ARIMA model with exogenous variables (SARIMAX). For each SARIMAX model there are seven parameters that are determined based on specific dataset properties. Those parameters are written as such: (p, d, q)(P, D, Q, s). As suggested by Ramos *et al*. (2015), these parameters are usually chosen based on AutoCorrelation and Partial AutoCorrelation functions (ACF and PACF) as well as Akaike Information Criterion (AIC). Having to perform these tests and manually choose the parameters is tedious, and the selections are only valid for the dataset that was used to perform the tests. Thus, the code was written to go through all possible parameter combinations and determine which combination results in the lowest AIC value, which is the estimated prediction error. According to Ramos *et al*. (2015), the model with the lowest AIC value is usually the best performing model, so this method of parameter selection works just as well as the traditional method. The *itertools* and *statsmodels* Python packages as well as the cited tutorial were used to accomplish this (Graves, 2020). The SARIMAX code was run for each data structure for both Customers and SKUs, which is a total of six times.

CNN Analysis

After splitting the dataset into train and test sets, the train set was detrended using a linear regression model. The prediction output the trend line for the dataset, and subsequently the trend value for each month was subtracted from the month's actual value. Figure 5 shows what the detrended dataset looks like for the total sales dollars (the variable that is predicted in this project), but the detrending was nonetheless completed for all variables.

After detrending the train set, the same data was deseasonalized, which involved subtracting from each value the value from the same month of the previous year. Of course, this meant that the train dataset was decreased from 48 months to 36 months because the first 12 months of data could not be deseasonalized. This left us with a 75/25 split for train/test – three years for train and one year for test. The ARIMA model still used four years of train data. It was decided that this is permissible because the project still answers the question of what each of these models can do when they start with five years of data.

After deseasonalization, the train dataset was standardized using *sklearn.preprocessing* StandardScaler and normalized using *sklearn.preprocessing* MinMaxScaler.

Table 1. The first five rows of Structure 1 for Customers, which includes all numerical variables as summations, averages, or the same as all individual Customers depending on the value. Values are exactly as they appear, not by the hundreds or thousands for example. The same structure was also applied to SKUs.

Invoice Month	Sales Dollars (USD)	Avg Sale Price of Each Item Purchased	Total Quantity Sold	Freight Dollars (USD)	Business Days Per Month	Number of Invoices
5/1/2017	819875.8677	495.3958326	2952767	33070.14926	22	1061
6/1/2017	866939.6777	495.963867	3471151	31838.72584	22	978
7/1/2017	697108.975	497.3265585	2623558	34911.9593	20	888
8/1/2017	920517.6011	486.1594091	3158789	36632.16485	23	1119
9/1/2017	885414.1115	483.3767757	3314323	50588.04821	20	1049

Table 2. The first five rows of Structure 2 for Customers, which includes all individual Customer sales as separate variables. Values are exactly as they appear, not by the hundreds or thousands for example. The same structure was also applied to SKU analyses.

Invoice Month	Sales Dollars (USD)	Cust101	Cust102	…	Cust194	Cust195
5/1/2017	819875.8677	11156.23131	540.3876		12189.7079	37404.3025
6/1/2017	866939.6777	14428.46415	222.2316		7698.4223	38238.9902
7/1/2017	697108.975	5649.367864	722.7891		10423.2471	38650.541
8/1/2017	920517.6011	12263.50017	778.467		7995.9907	55018.2972
9/1/2017	885414.1115	12639.24433	567.7623		7068.0297	41367.304

Once preprocessing was complete, the *keras* package from Python was used to create a multi-step, multichannel CNN model. This means the model could take multiple inputs and predict multiple months at a time. In this case, the model used the variables in Tables 1 and 2 as inputs and outputted 12 months of predicted sales dollars (USD). The method for multi-step time series forecasting described in the cited tutorial was used along with the *keras* package in Python to create the multichannel CNN model with a few slight changes (Brownlee, 2020). Because this project forecasted a year of sales, the model split the data into years rather than weekly groupings. In addition, the batch size was changed to 12 to better accommodate the input data, and the activation function was changed to 'sigmoid.' The reason for the change in activation function is that 'sigmoid' is the most common activation function and it most closely represents biological neurons in the human brain (Pan and Zhou, 2020). It also produced better results than the original 'reLU' function that was used in the tutorial.

Because the CNN model is stochastic, a different prediction is created each time it is run. Thus, the model was run 10 times for each data structure for Customers and SKUs. Then the average scores for each of the 10 runs was recorded.

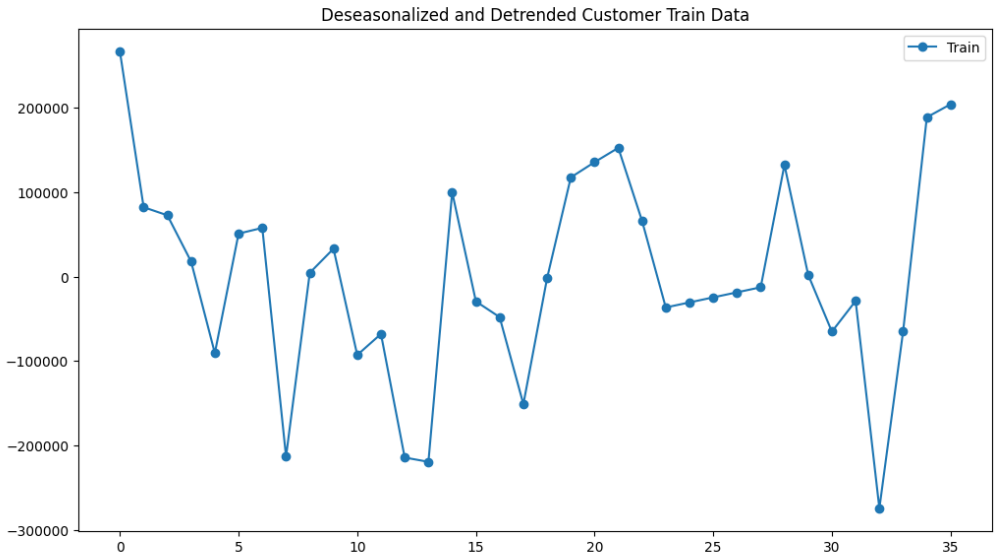

Figure 5. The train sales data (USD) for the pooled Customer CNN analysis after deseasonalization and detrending. There is much less variability from month to month (36 months total, May 2018 through April 2021) after deseasonalization, and detrending rendered the trend line for the timeseries completely flat at y=0.

Results

Results for each forecasting model were recorded using three different performance measures, and various graphs were also created to visualize the predictions in contrast with the test data points.

Performance Measures

As is standard practice for forecasting models, Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) scores were used to evaluate model performance. These were calculated using the *sklearn.metrics* Python package. See Table 3 for equations. Table 3 provides a direct breakdown of metrics and the formulas used in RMSE, MAE, and MAPE. Each equation uses values based on month and time values.

The MAPE metric allows for direct comparison of models with different datasets, and it is generally accepted that models with a MAPE score below 10% are very good. RMSE and MAE are dependent on the dataset, so they can only be used to compare the performance of different data structures using the same dataset (in this case either Customers or SKUs). Minimum, maximum, and average values of the test dataset can be useful for interpreting RMSE and MAE scores, so they are noted in Table 4.

Table 3. Equations for performance metrics RMSE, MAE, and MAPE (Temür *et al.*, 2019). n refers to the number of months in the test data, y_t to the actual value at time t, and $\hat{y}_t$ the predicted value at time t.

Metric	Formula
RMSE	$RMSE = \frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2$
MAE	$MAE = \left(\sum_{t=1}^{n}\left\|\frac{y_t - \hat{y}_t}{n}\right\|\right)$
MAPE	$MAPE = \left(\sum_{t=1}^{n}\left\|\frac{y_t - \hat{y}_t}{n}\right\|\right)\left(\frac{100}{n}\right)$

Table 4. Minimum, maximum, and average sales dollars (USD) for the Customer and SKU test datasets. These are useful for interpreting RMSE and MAE scores.

	Customer	SKU
Test Min	1050116.737	129159.157
Test Max	1606915.652	179904.430
Test Avg	1305323.228	155211.178

Resulting Scores

All scores for each model, dataset, and data structure are denoted in Table 5. Each model and data combination has an RMSE, MAE, and MAPE value. Additionally, the SARIMAX models have an AIC score and a column for which parameters were used for the model. Each SARIMAX model used a different set of parameters according to whichever set out of all combinations resulted in the lowest AIC value.

SARIMAX has generally outperformed CNN when looking at the MAPE scores, except in the case of Structure 1, which performed similarly to CNN for Customers and much worse than CNN for SKUs. Given the high AIC scores associated with the parameters used for Structure 1 in the SARIMAX models, this is not surprising. All other SARIMAX models returned MAPE scores between 2 and 4.5% and would be considered good forecasts, while all the CNN models returned MAPE scores over 12%.

Out of all Customer SARIMAX models, Structure 2 performed the best, while Structure 3 performed the best out of all SKU SARIMAX models. Structure 1 performed the best out of all Customer CNN models by a small margin, and Structure 2 won by a small margin for SKU CNN models.

Figures 6 and 7 show the predicted values in comparison to the test values for the best Customer SARIMAX and CNN models, respectively. The best Customer SARIMAX model used Structure 2, and the best Customer CNN model used Structure 1. Figures 8 and 9 show the predicted values in comparison to the test values for the best SKU SARIMAX and CNN models, respectively. The best SKU SARIMAX model used Structure 3, and the best Customer CNN model used Structure 2.

Table 5. The results of the SARIMAX and CNN models for both Customer and SKU datasets and the three data structures as described in the text.

Model Type	Dataset	Data Structure	AIC	Parameters	RMSE	MAE	MAPE
SARIMAX	Customer	3	-558.485	(1,1,2)(0,1,1,12)	59093.813	51678.824	0.04002
SARIMAX	Customer	2	-543.195	(0,1,1)(1,1,1,12)	38275.895	29184.995	0.02318
SARIMAX	Customer	1	888.000	(0,1,1)(0,1,1,12)	176792.524	151685.065	0.11324
SARIMAX	SKU	3	-583.675	(2,1,2)(1,1,1,12)	5474.281	4440.358	0.02869
SARIMAX	SKU	2	-596.597	(2,1,1)(1,1,1,12)	7939.755	6865.875	0.04429
SARIMAX	SKU	1	743.555	(2,1,2)(2,1,1,12)	41775.123	37453.611	0.23383
CNN	Customer	3			202287.611	166264.611	0.12319
CNN	Customer	2			211940.638	165132.059	0.12098
CNN	Customer	1			207118.599	164138.459	0.12056
CNN	SKU	3			23904.587	19553.745	0.12323
CNN	SKU	2			24092.627	19569.444	0.12313
CNN	SKU	1			24157.760	19778.419	0.12461

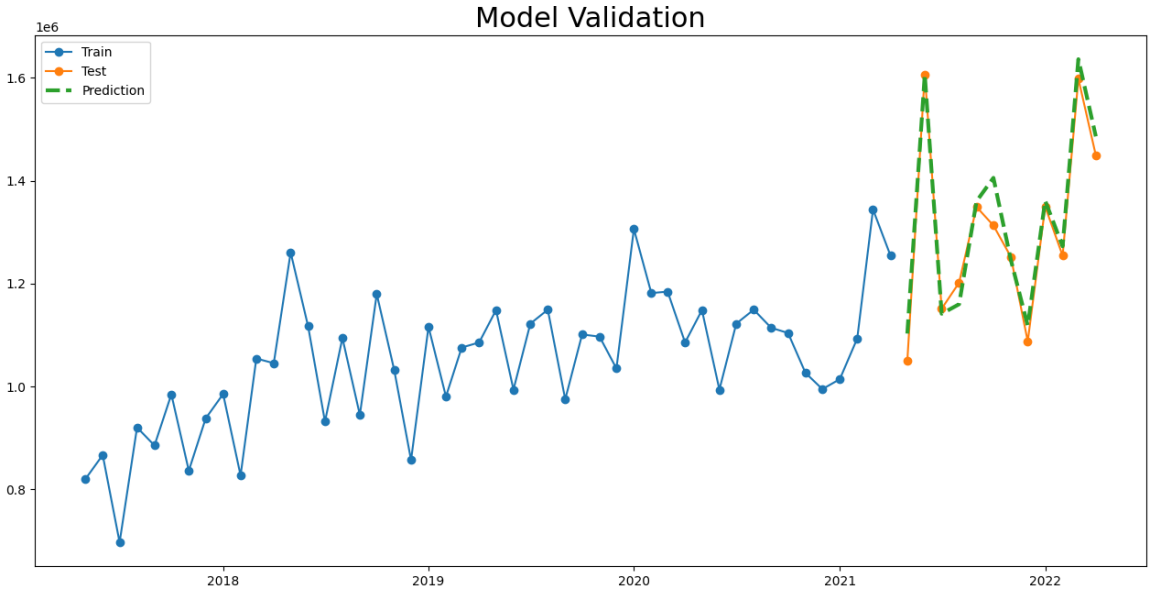

Figure 6. A comparison of test and prediction points (Sales USD) for the most accurate Customer SARIMAX model which has a MAPE score of 2.318%. This model used Structure 2, which includes a variable for the sales of each of the 95 customers included in this project.

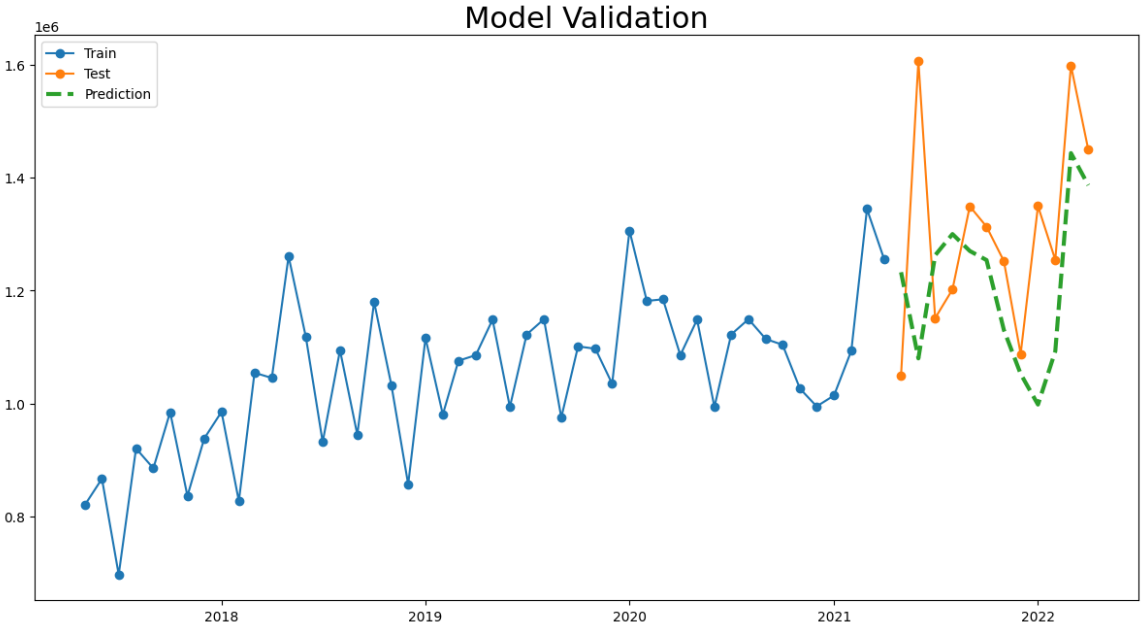

Figure 7. A comparison of test and prediction points (Sales USD) for the most accurate Customer CNN model which has an average MAPE score of 12.056%. This model used Structure 1, which uses five numerical variables associated with all customers included in this project. Even though the model only took three years of training data, May 2017 to April 2018 was still included in this graph because those years were required to complete deseasonalization and thus contributed to the final forecast.

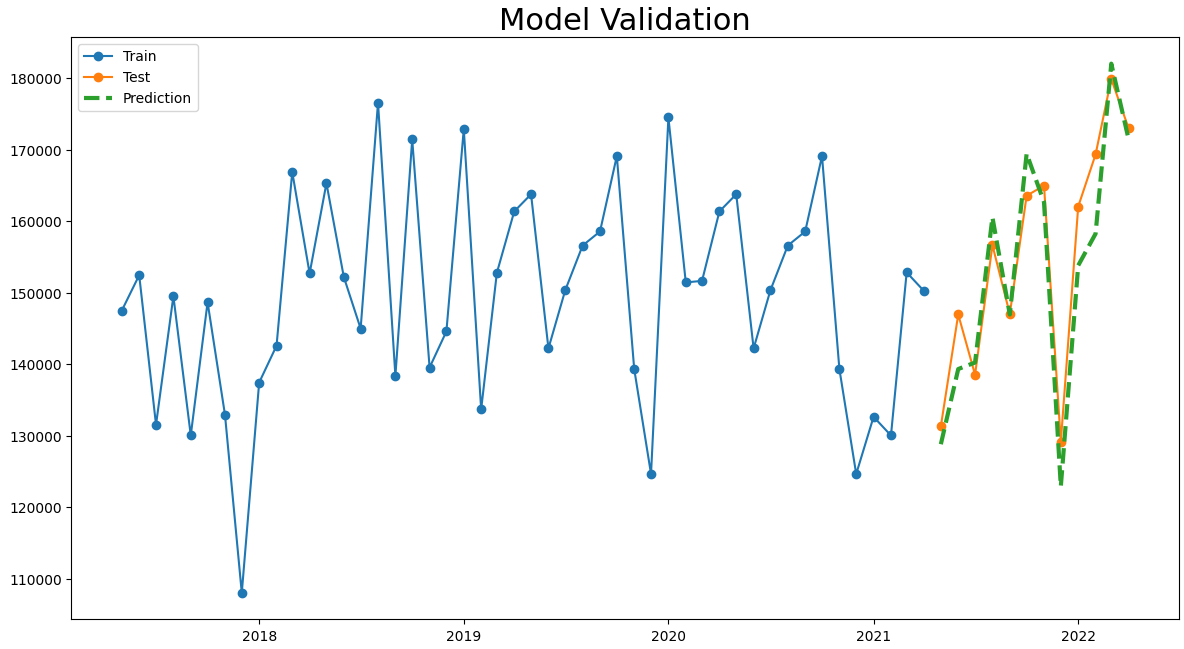

Figure 8. A comparison of test and prediction points (Sales USD) for the most accurate SKU SARIMAX model which has a MAPE score of 2.869%. This model used Structure 3, which includes one variable for the sales of each SKU included in the project and five numerical variables associated with all SKUs.

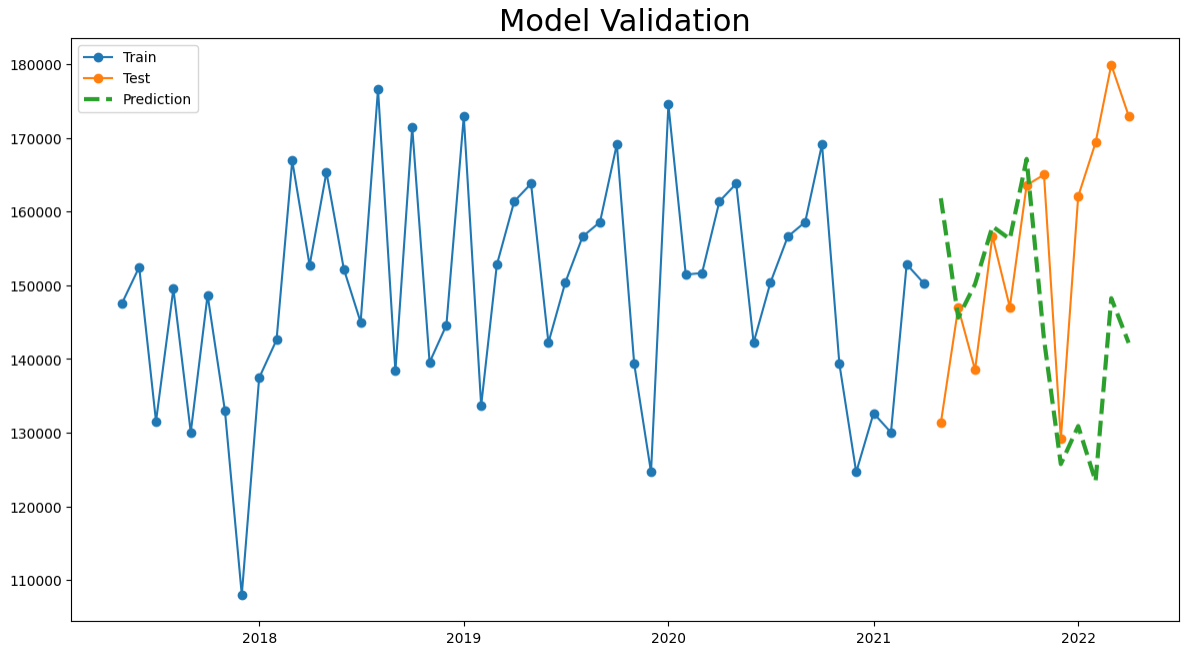

Figure 9. A comparison of test and prediction points (Sales USD) for the most accurate SKU CNN model which has an average MAPE score of 12.313%. This model used Structure 2, which includes a variable for the sales of each of the SKUs included in this project. Even though the model only took three years of training data, May 2017 to April 2018 was still included in this graph because those years were required to complete deseasonalization and thus contributed to the final forecast.

Discussion

SARIMAX performed better than CNN in almost all instances, but there were several limitations in this project that may have affected the outcome, both in terms of the data and models used.

Data

Results suggest that the COVID-19 pandemic data has had some effect on the forecasts even with the modifications conducted during data preprocessing. This is not surprising given that the cutoff between the train and the test data for both Customers and SKUs is April-May 2021, which is just about the time when the economy began to recover from the pandemic, at least for Company A. As shown in Figure 3, the train data for Customers had a slightly positive trend through April 2021, and then it boomed going forward. The SKU train data even saw a slightly downward trend, whereas the test data saw a positive trend (Figure 4). Even with the modifications to the train data to account for the pandemic, there was still an effect, although despite the evidence proposed it remains a mystery whether this effect was directly because of COVID-19 or if there may have been other factors at play. Perhaps in the next year when the current test data will become part of the train dataset the effect on performance will be much smaller, and a better determination can be made regarding the origin of this error.

In conjunction with the pandemic impacting the performance of the CNN models, the amount of data may have also posed some issues. As discussed previously, Chu and Zhang (2003) suggest that five years of data is sufficient for forecasting purposes. However, that project used daily data, whereas this project used data aggregated by month, resulting in far fewer data points than were used by Chu and Zhang (2003). In addition, the deseasonalization of the data reduced those five years down to four for the CNN models. Because of this, it would be prudent to include more data points in CNN models moving forward. In contrast, the SARIMAX models performed very well despite the lower volume of data. In a case such as this where more than five years of data is not easily accessible, it may be better to use SARIMAX over CNN.

Another limitation of this project in terms of data was that only numerical variables were used. It would be prudent in future studies to incorporate categorical variables into the CNN or other machine learning models to increase performance.

Models

The results make the SARIMAX models appear more reactive to differences in data structures and variables than CNN, which was relatively consistent in performance across all data structures. This observation cannot be generalized to all SARIMAX and CNN models, however, because each SARIMAX model used the optimal parameters for a given dataset whereas CNN does not have parameters that can be optimized in a similar manner. The CNN parameters such as filters, batches, kernel size, layers, etcetera can be optimized, but this is more of a “guess and check” operation than a definitive assessment of which parameters result in the best outcome. This optimization of parameters could explain the greater variability in performance for the SARIMAX models, and it reiterates the importance of testing and optimizing models for individual datasets as discussed by Caglayan *et al*. (2020).

One way that the CNN model could be more optimized for the given data is by decreasing the prediction window. Since SARIMAX performed well as is, 12 months seems to be fine in that case. However, looking at Figure 9, the prediction points for the SKU CNN models diverge from the test points starting at seven months of predictions, so it is possible that predicting just three to six months at a time could improve performance.

The results of a study by Ou, Chen, and Tsai (2020) conducted on sales forecasting for convenience stores found that sales were highly variable from region to region. Because of this, future studies should group Customers and SKUs by region when conducting pooled forecasts rather than using simple randomization as this project did. By extension, it could also be better to group Customers and SKUs by industry for more accurate results. Customers and SKUs from the same industry may see very similar seasonality and trends, which may make it easier for any model, but especially machine learning models, to detect those patterns.

It may also be useful for companies to have the ability to forecast individual Customers or SKUs rather than making forecasts for a pool of Customers or SKUs. Because of this, pooled analyses for SARIMAX and CNN models could also be compared to individual analyses in the future. According to Ma and Fildes (2021), modeling time series individually considers each time series' own characteristics, but it cannot capture common patterns across different time series. When performing retail sales forecasts at the SKU level, they found that machine learning methods often work better than statistical methods when the data is pooled, but it is best to keep things simple using statistical methods when making individual forecasts (Ma and Fildes, 2021). This indicates that there may be potential with individual forecasts using SARIMAX, but it may also be worthwhile to evaluate individual forecasts using CNN.

Many studies have also used hybrid methods to forecast sales and have found success. Menculini *et al*. (2021) produced one such study, which found a hybrid model of LSTM and CNN capabilities produced the best results for food sales predictions. Temür *et al.* (2019) also used a hybrid model in their study of housing sales that combined capabilities of LSTM and ARIMA, and they found that the hybrid model performed better than individual LSTM and ARIMA models. Given these examples, hybrid models may be worth exploring in future work. However, complex models often require more training time and resources, which may not be worth the improved performance in many cases (Menculini *et al*., 2021).

In addition to testing out hybrid models, other machine learning models could be tested in the future. There are many different types, such as Neural-multilayer perceptron (MLP), Random Forest (RF), and Support Vector Machine (SVM) to name a few (Ou *et al*., 2020). Perhaps one of these used for the Customers and SKUs of Company A could outperform CNN or even SARIMAX.

Conclusions

The purpose of this project was to compare the performance of statistical and machine learning models for Company A data and determine which is best for use under the given circumstances. This project looked at one model from each category, SARIMAX as a statistical model

and CNN as a machine learning model.

The results show that SARIMAX performs much better than CNN at this juncture, returning smaller RMSE, MAE, and MAPE values in almost all circumstances. SARIMAX is also much easier to use and understand, making it the optimal forecast modeling option for Company A.

There were, of course, several limitations presented in this project. These included access to only years of data, the inclusion of only numerical variables for both models, the use of simple randomization rather than strategic pooling for data selection, the effect of the COVID-19 pandemic on the overall trend and seasonality of the time series for both Customers and SKUs, and the use of single models rather than testing out hybrid options. Reworking this project in such a way that most of those limitations are removed may lead to different results, but for Company A specifically it is likely best to move forward with SARIMAX as it performs well, and it is relatively easy to use when compared with more complex models such as CNN.

Acknowledgements

Data for this project was supplied by the anonymous Company A. Feedback and encouragement was provided by Saint Mary's University of Minnesota professors John Ebert and Greta Poser as well as students Annika Feight, Per Lundmark, and Ian Robertson.

References

Brownlee, J. 2020. Convolutional Neural Networks for multi-step time series forecasting. Machine Learning Mastery. Retrieved May 22, 20222 from https://machinelearningmastery.com/how-to-develop-convolutional-neural-networks-for-multi-step-time-series-forecasting/?unapproved=671286&moderation-hash=fd1590e23678352074a2a52b1fbc3b24#comment-671286.

Caglayan, N., Satoglu, S.I., and Kapukaya, E. 2020. Sales forecasting by artificial neural networks for the apparel retail chain stores-An application. Journal of Intelligent & Fuzzy Systems, 39(5), 6517-6528. Retrieved Jan 21, 2022 from Business Source Premier database.

Chu, C.W., and Zhang, G.P. 2003. A comparative study of linear and nonlinear models for aggregate retail sales forecasting. International Journal of production economics 86(3), 217-231. Retrieved Jan 21, 2022 from ScienceDirect database.

Fabianová, J., Kačmáry, P., Molnar, V., and Michalik, P. 2016. Using a Software Tool in Forecasting: a Case Study of Sales Forecasting Taking into Account Data Uncertainty. Open Engineering, 6(1). Retrieved Jan 21, 2022 from DOAJ database.

Graves, A. 2020. Time Series Forecasting with a Sarima model. Medium. Retrieved May 22, 2022 from https://towardsdatascience.com/time-series-forecasting-with-a-sarima-model-db051b7ae459.

Lalou, P., Ponis, S.T. and Efthymiou, O.K. 2020. Demand forecasting of retail sales using data analytics and statistical programming. Management & Marketing. Challenges for the Knowledge Society, 15(2), 186-202. Retrieved Jan 21, 2022 from DOAJ database.

Ma, S., and Fildes, R. 2021. Retail sales forecasting with meta-learning. European Journal of Operational Research, 288(1), 111-128.

Retrieved Jan 21, 2022 from ScienceDirect database.
Menculini, L., Marinia, A., Proietti, M., Garinei, A., Bozza, A., Moretti, C., and Marconi, M. 2021. Comparing Prophet and Deep Learning to ARIMA in Forecasting Wholesale Food Prices. *Forecasting, 3*(40), 644-662. Retrieved May 22, 2022 from DOAJ database.
Ou, T.Y., Chen, Y.J., and Tsai, W.L. 2020. Sales Forecasting of Perishable Foods With Multiple Stores and Communities-An Empirical Study of Convenience Stores in Taiwan. *International Journal of Intelligent Technologies and Applied Statistics, 13*(4), 385-409. Retrieved Jan 21, 2022 from Academic Search Premier database.
Pan, H., and Zhou, H. 2020. Study on convolutional neural network and its application in data mining and sales forecasting for E-commerce. Electronic Commerce Research, 20(2), 297-320. Retrieved Jan 26, 2022 from Business Source Premier database.
Ramos, P., Santos, N., and Rebelo, R. 2015. Performance of state space and ARIMA models for consumer retail sales forecasting. *Robotics and computer-integrated manufacturing, 34*, 151-163. Retrieved Jan 25, 2022 from ScienceDirect database.
Temür, A.S., Akgün, M., and Temür, G. 2019. Predicting housing sales in Turkey using ARIMA, LSTM and hybrid models. Retrieved Jan 25, 2022 from DOAJ database.