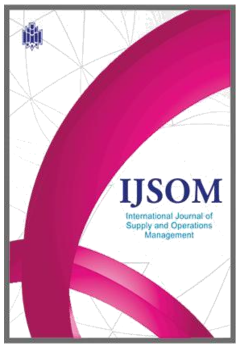
International Journal of Supply and Operations Management

IJSOM

2024, Volume 11, Issue 4, pp. 390-416
ISSN-Print: 2383-1359
ISSN-Online: 2383-2525
www.ijsom.com



Multivariate Sales Forecasting Using Gated Recurrent Unit Network Model

W.A. Roshan S. Jayasekara [a], P. T. Ranil S. Sugathadasa [a], Oshadhi K. Herath [a,c] and H. Niles Perera [a,b*]

[a] *Department of Transport Management and Logistics Engineering, University of Moratuwa, Sri Lanka*
[b] *Center for Supply Chain, Operations and Logistics Optimization, University of Moratuwa, Sri Lanka*
[c] *Extreme Energy-Density Research Institute, Nagaoka University of Technology, Nagaoka, Japan*

Abstract

Market forecasting is an integral part of supply chain management. Machine learning models have turned a new page in predictive analysis and helped organizations achieve improved accuracy. This paper focuses on creating a Gated Recurrent Unit (GRU) model to predict sales for multiple stores as a multivariate time series. GRUs are a variation of Recurrent Neural Networks (RNNs) used to sequence modelling tasks. The dataset used to create the model contains the unit sales of 3,049 SKUs sold in 10 stores. The sales data from the 3049 SKUs were grouped into the 7 departments to use as input to the model. A Vector Autoregression (VAR) and LightGBM models were used to compare the GRU model predictions. Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE) were used to compare the 2 models. The mean MAPE values for forecasts of the GRU, VAR, and LightGBM models were 13.77%, 14.87%, and 14.24% respectively, while MAE values were 68 Units, 72 Units, and 69 Units Respectively. The study reveals that the GRU model provides more accuracy for multivariate sales forecasting due to its ability to learn hidden patterns automatically and handle time mechanisms such as trends and seasonality.

Keywords: Multivariate Sales Forecasting; Deep Learning; Recurrent Neural Networks (RNN); Supply Chains; Gated Recurrent Unit (GRU).

1. Introduction

Market forecasting is an integral part of supply chain management. Forecasting is the root of all supply chain decisions. Although demand forecasting is undeniably significant, it is also one of the most challenging facets of supply chain planning. Market Forecasting supports essential industry practices such as budgeting, financial decision-making, distribution, marketing strategies, raw material purchasing, development planning, risk management, and reduction strategies. Hence, organizations are trying to develop a highly accurate model using past demand data (Forslund & Jonsson, 2007). Demand forecasting can be done using both Quantitative and qualitative approaches. However, the industry's academic and standard practices suggest that demand forecasts are often subject to human activity (Perera et al., 2019). Organizations spend their resources on supply chain projects that develop forecasting models to improve the functionality of the supply chain (Arachchige et al., 2021).

Over the last decade, the emergence of deep learning has revolutionized the industry as the motivational factor behind all possible machine learning benchmarks. Machine vision and natural language processing are some of the popular applications in the field of Artificial Neural Networks (ANN). This new technology can capture the hidden data

*Corresponding author email address: hniles@uom.lk
DOI: 10.22034/ijsom.2024.109038.2141

patterns used to make predictions. The capability of ANNs has realigned the possible future of sales forecasting (Zhang et al., 1998).

Significant work has been conducted in forecasting and inventory management, including creating modern statistical distributions for overall lead-time demand and several enhanced variants of Croston's method for predicting irregular time sequences (Gardner, 2006). Modelling of time series is a dynamic research field that has continued to grow over the past few decades (Hewamalage et al., 2020). The main objective of time series modelling is to gather and thoroughly analyze historical data to create a fitting model that explains the series' intrinsic structure and to establish the temporal ordering among certain variables (Wang et al., 2024). Then, this model is used to produce potential values to continue the sequence and make future predictions. Time series forecasting can thus be characterized by considering the past as predicting the future (Azubuike & Kosemoni, 2017).

Moving average models are the simplest yet most common forecasting methods in the time series analysis. A Moving average model takes N measurements' most recent numerical average, where N is a given number based on the expected data type (Cochran et al., 2011). Autoregressive Integrated Moving Average (ARIMA) is one of the most widely used standard methodologies, a generalization of the Autoregressive Moving Average System (Gilbert, 2005). ARIMA models have a significant divergence from predicting high-frequency financial time series, suggesting that there is still potential for development(Li et al., 2020). High computing power and modern database architecture allow data to be stored and processed with finer and finer pellets to generate lower and lower count data time series. The estimated methods sufficient for continuous probability distributions can no longer be extended to these sequences (Kolassa, 2016). Because of that, machine learning models like Random Forest Regression and XGBoost have also been used to obtain better results using non-linear data (Helmini et al., 2019).

In several fields, such as stock market forecasting, weather forecasting, complex dynamic system analysis, and Internet of Things data analysis, multivariate time series forecasting has become widely used (Ahaggach et al., 2024). Provided multiple time series in which any or all of them are associated to some degree, a significant challenge has been exploring and manipulating the complexities and correlations between them while making forecasts in a reasonable period (Chang et al., 2018). Over the years, multivariate time series forecasting was embraced, possibly because of its capacity to bring other variables along. However, the critical objective of modelling the time series is estimation, and a successful model can forecast the actual data values with fewer errors. Naturally, if a model can predict the actual data values well, it is apparent that the predictions are accurate (Azubuike and Kosemoni 2017).

Linear and non-linear time series models have limited predictive potential in the presence of nonlinearity and non-normality, where the functional structure of the interaction between inputs and outputs is uncertain. When the relevant characteristics are present, machine learning models outperform univariate techniques in identifying the post-promotion impact(Hewage & Perera, 2021). Artificial Neural Networks are used for this purpose(Šestanović & Arnerić, 2021). Neural networks are very complex models for non-linear simulation, which can be used to forecast time series. The equations' parameters are modified using a process of optimization. Various neural networks have different computational equations and various methods of optimization. Optimization approaches vary from basic techniques like gradient descent to more complex procedures like genetic algorithms (Bandara et al., 2019). ANNs work like a system of interconnected neurons, which is how the human brain works. Through an objective function, primarily a loss function, ANNs can calculate a collection of values as inputs, generating the desired output. Neural networks differentiate themselves by not defining all parameters specified by the author.

Computing these parameters is done by exposing the network to several thousand instances and changing internal parameters to make sure the maximum performance is reached. The findings of the ANN demonstrate the potential of ANN to manage complicated data, including short and seasonal time series, beyond expectations and suggest several avenues for future study (Crone et al., 2011). Due to the relatively short nature of most time series, ANNs (and other strongly non-linear and nonparametric methods) are unsuitable for univariable time series predictions (Hyndman, 2020). The most typical Deep Learning architectures for sales forecasting are LSTM, Deep neural networks, and Multi-layer perceptron (Chen et al., 2023). Deep learning prediction models enable new capabilities that conventional models may not be able to achieve (Eglite & Birzniece, 2022).

Despite their strength, there are limitations to regular neural networks. They are based on the premise that preparation and test examples are separate. The whole state of the network is lost after each instance is processed. However, it is inappropriate where data points are connected in time or space. Standard networks often depend on examples of fixed length vectors. This means extending these powerful learning instruments for modelling data with temporal or sequential structures and varying inputs and outputs, particularly in many already state-of-the-art fields with Recurrent Neural Networks (RNNs). RNNs are connection models with the capacity of cross-sequence data selectively transferring information when processing sequential data from one part to another (Lipton et al., 2015). RNN is a type of Artificial Neural Network used for sequential data (Predić et al., 2024). In standard ANNs, inputs and outputs are independent, but in RNNs, all the information is connected.

When teaching itself, the RNN remembers all these links. To do this, the RNN builds networks with loops, allowing the knowledge to remain. The ability to create and modify complex RNN models may not apply to too many users who are used to conventional time series techniques. They would like to use familiar and easier-to-interpret and develop strategies. Thus, they might be hesitant to use RNNs to achieve adequate precision despite the recent achievements of RNNs in forecasting. This is also closely linked to the current disagreement in the forecasting world over whether so-called off-the-shelf learning strategies will surpass classical benchmarks. Besides the intuitions described above regarding small single series against more extensive multivariate time series, there are no known rules for when conventional statistical methods are better at modelling time series than RNNs (Hewamalage et al., 2020). Long Short-Term Memory(LSTM) networks are among the more popular RNN models for sales forecasting. LSTMs predict sales accurately and automatically. This sheds light on the potential for sophisticated neural network approaches to improve sales forecasting methodologies(He et al., 2022).

GRU is a specific form of RNN. GRU neural networks are among the most effective and efficient ANNs. GRU networks are explicitly designed to prevent the inconvenience of long-term reliance. GRU neural networks outperformed the LSTM on most tasks (Qin et al., n.d.). GRU performs a similar scheme for any part of a chain. That is the reason it is referred to as recurring. The outcome was based on previous equations. Another theory concerning GRUs is that they have a device called a memory unit. The knowledge determined is captured by it. GRUs also allow information to be transmitted over time stages (Obaidur Rahman et al., 2019).

2. Problem Definition

A crucial component of supply chain management is market forecasting, which enables businesses to match customer demand and make well-informed plans precisely. Precise sales forecasting is essential in reducing inventory expenses, guaranteeing product accessibility, and increasing operational effectiveness. To tackle this obstacle, establishments have resorted to machine learning models, transforming predictive analysis by offering sophisticated instruments for using past demand information.

The main topic of this article is the creation and assessment of a Gated Recurrent Unit (GRU) network model for multivariate sales forecasting. For modern enterprises, multivariate sales forecasting is an essential andchallenging process that involves several variables and time series data. Recurrent neural networks (RNNs) have shown great promise in modelling sequential data, and one particular RNN variation that performs well for capturing temporal dependencies in time series data is the GRU. To address the multivariate character of the problem, the model we present attempts to predict sales for several stores spanning a varied range of stock-keeping units (SKUs). The study's dataset includes unit sales information for 3,049 SKUs from ten shops. To make the most of this data, we have divided the 3,049 SKUs' sales records into seven different departments. As part of our study technique, we compare the GRU model's performance to that of two alternative models: the LightGBM model and the Vector Autoregression (VAR) model. A comparative analysis is necessary to evaluate if the GRU model is better at capturing the complexities of multivariate sales forecasting.

The results of this study should show that the Gated Recurrent Unit (GRU) model performs exceptionally well in multivariate sales forecasting because of its innate ability to discover hidden patterns automatically and to handle time-related complications like trends and seasonality. The research's findings have significant ramifications for businesses looking to increase the precision and effectiveness of their sales forecasting procedures.

3. Methodology

This segment discusses the design of the research. As the study focuses on creating a GRU network model, LIghtGBM, and a VAR model to estimate sales demand, historical sales data were obtained from secondary data sources. This section focused on research methodology, including research design, data collection, model development of the GRU model, and the VAR model describing the associated steps. Model evaluation matrices are also discussed in the latter part. The novelty of the research is building a GRU model to predict multivariate sales data and further comparing it with the VAR and LightGBM models.

3.1 Data

Time series data of past sales were obtained from the dataset of the M5 forecasting challenge hosted by the Kaggle community, which ran from 2 March to 30 June 2020. The dataset contains sales data for ten stores of Walmart, an American multinational supermarket chain, encompassing seven departments (product families). Each department would have several stock-keeping units (SKUs) assigned to them. The dataset includes the unit sales of various products arranged as clustered time series. The dataset contains the sales of 3049 items, which can be categorized into seven other departments.

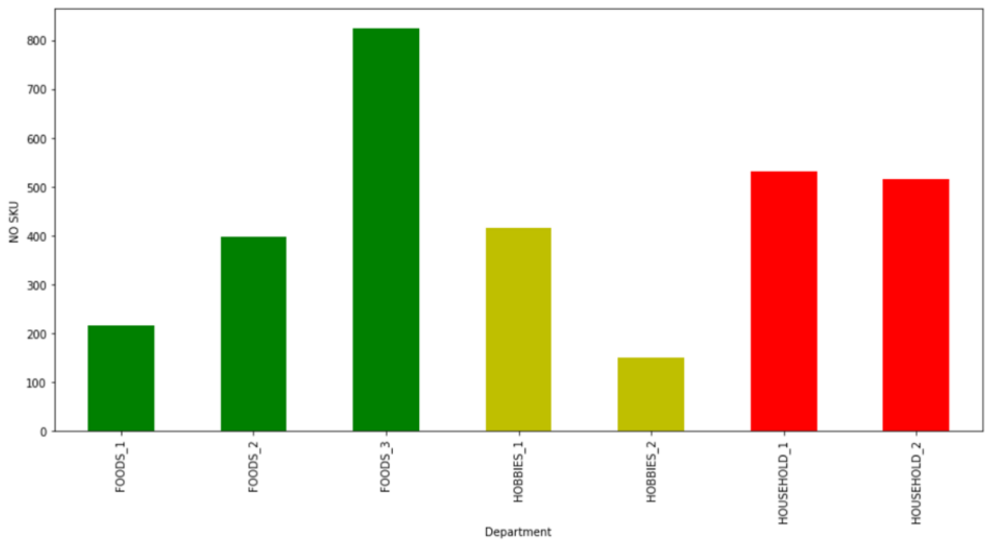


Figure 1. SKU vs department plot

Observations are not independent during time series analysis; hence, splitting the data as in the case of non-time series analysis does not work because, in non-time series analysis, training and testing data will be randomly divided into sets. Therefore, time series observations are usually broken along with the sequences. Sales data for one month (30 days) was selected from the end of the time series to test and compare the model. One thousand nine hundred twelve days (98% of the time series) were used to train the model.

3.2 GRU and LightGBM Model Development

Figure 2 shows the steps in building GRU and LightGBM models to forecast demand. The model was developed using the Python programming language. Keras library with TensorFlow is used to create and train the neural network models, and for analysis and preprocessing pandas, Scikit-learn packages were used.

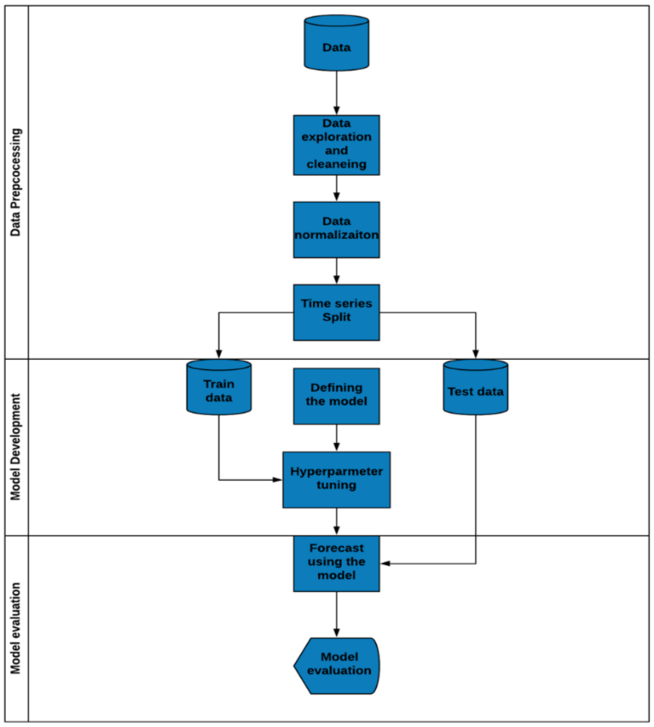


Figure 2. GRU Model development workflow

Data processing entails normalization and standardization before training a neural network model to rescale input and output variables (Lai, 2006). Each of the time series was scaled using the following equation in the formula $\bar{x}$ is the mean of the time series and σ is the standard deviation.

$$Z = (x - \bar{x})/\sigma \tag{1}$$

The sequential model is used to construct the models using the Keras library. It enables a model layer by layer to be built. Each layer has weights that fit the following layer. The input layer is a GRU layer followed by two hidden layers, which are also GRU layers, and in between hidden layers, dropout layers were positioned, which helped to reduce overfitting during the training process. Finally, a dense layer is set up to get the output.

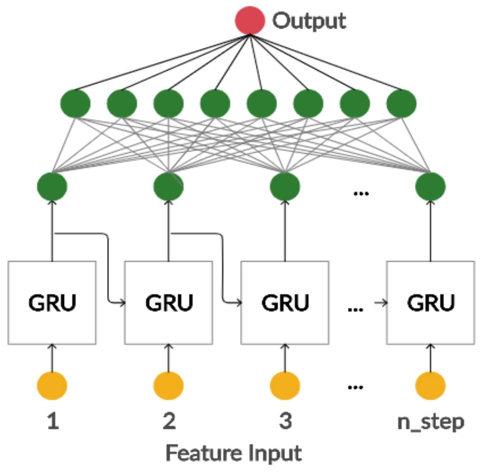


Figure 3. GRU Model Architecture

A GRU layer should always be in 3 dimensions. The first dimension is the batch size, which is the number of samples that need to be processed before updating the neural network, and the next dimension is the time step, which represents the number of days that were used to predict the next day's forecast. The other dimension is the considered features, the number of variables in the multivariate time series data frame. Features are the sales of different departments in each store that add up to 70-time series.

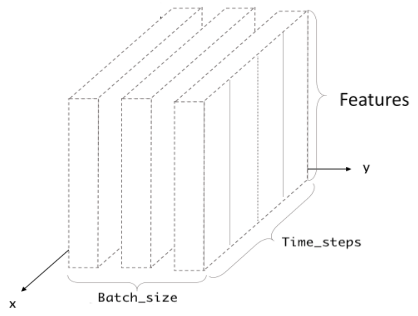


Figure 4. GRU input shape

The loss function used in the study is a mean squared error (MSE). MSE loss is the cumulative mean of the square differences between actual and expected values. MSE is sensitive to outliers, and their cumulative target value would be the best prediction, provided multiple instances of the same input function values. The formula for the MSE is defined as follows.

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

Hyperparameter tuning uses the Bayesian optimization technique since manually tuning hyperparameters is inefficient. Bayesian optimization offers a Bayes theorem-based principled technology for an efficient and effective global optimization problem. The GRU and LightGBM models consist of several hyperparameters that can be tuned to get the best model. Table 1. Hyperparameters and search spacesshows the hyperparameters that were used in the research. LightGBM model hyperparameters and search spaces are shown in Table 2.

Table 1. Hyperparameters and search spaces for the GRU model

Hyperparameter	*Search space*
Layer 1 units	Integer (50,400)
Layer 2 units	Integer (50,400)
Layer 3 units	Integer (50,400)
Layer 1 activation	Categorical (['relu', 'tanh'])
Layer 2 activation	Categorical (['relu', 'tanh'])
Layer 3 activation	Categorical (['relu', 'tanh'])
Batch size	Integer (2,256)
Epochs	Integer (5,100)
Dropout rate	Real (0.1,0.8)

Table 2. Hyperparameters and search spaces for the LightGBM model

Hyperparameter	*Search space*
max_depth	Integer (50,500)
num_leaves	Integer (50,500)
min_data_in_leaf	Integer (50,500)
n_estimators	Integer (50,500)

3.3 VAR Model Development

The VAR model was developed to compare the results of the GRU model forecast. The VAR model was also developed using Python, and the steps involved are shown in Figure 5. In an autoregression model, a linear combination of past values of a variable is used to predict the same variable's future values. The word autoregression reveals that it is a regression against itself of the variable. The VAR model is an extension of the autoregression that uses multiple variables. VAR models are distinguished by their order, corresponding to the number of prior periods used by the model, also known as the lag length (p).

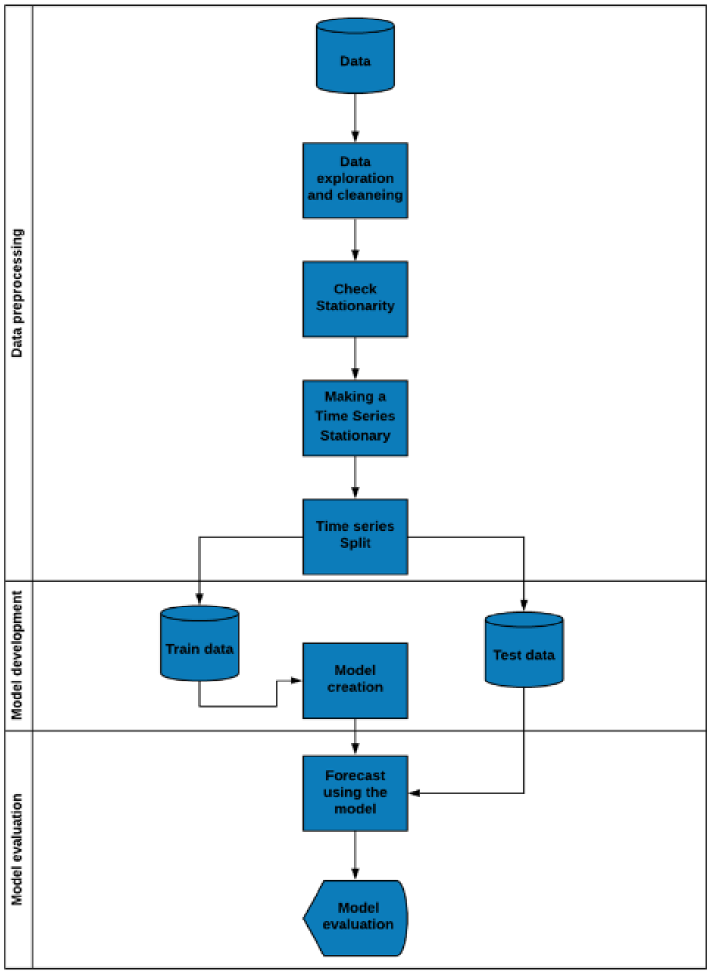


Figure 5. VAR model development workflow

Checking the stationarity of the time series is important when modelling the time series using the VAR model. Stationary time series correlation coefficients will be more consistent than non-stationary correlation coefficients and would also impact the output of subsequent data processing steps. Observations from a non-stationary time series show seasonal impacts, patterns, and other mechanisms that depend on the time index (Yang & Shahabi, 2005). Stationarity is checked using the Augmented Dicky Fuller (ADF) test.

Table 3. Stationarity check Results (Augmented Dicky fuller test)

series	*p_stat*	*Status*	*p_stat_differenced*	*Status_differenced*
CA_1_FOODS_1	0.004311053	Stationary	2.72E-26	Stationary
CA_1_FOODS_2	8.16E-05	Stationary	0	Stationary
CA_1_FOODS_3	0.148980919	Non-Stationary	0	Stationary
CA_1_HOBBIES_1	0.019833431	Stationary	5.30E-27	Stationary
CA_1_HOBBIES_2	0.01739633	Stationary	8.46E-24	Stationary
CA_1_HOUSEHOLD_1	0.532348411	Non-Stationary	1.43E-27	Stationary
CA_1_HOUSEHOLD_2	0.130899043	Non-Stationary	2.54E-25	Stationary
CA_2_FOODS_1	0.006678353	Stationary	2.96E-24	Stationary
CA_2_FOODS_2	0.99026148	Non-Stationary	1.65E-27	Stationary
CA_2_FOODS_3	0.763985019	Non-Stationary	5.43E-28	Stationary
CA_2_HOBBIES_1	0.062681006	Non-Stationary	3.08E-27	Stationary
CA_2_HOBBIES_2	0.011737301	Stationary	3.89E-27	Stationary
CA_2_HOUSEHOLD_1	0.324011227	Non-Stationary	1.65E-27	Stationary
CA_2_HOUSEHOLD_2	0.269248565	Non-Stationary	2.08E-26	Stationary
CA_3_FOODS_1	0.014083316	Stationary	1.08E-28	Stationary
CA_3_FOODS_2	0.001077103	Stationary	0	Stationary
CA_3_FOODS_3	0.107024364	Non-Stationary	0	Stationary
CA_3_HOBBIES_1	0.190229232	Non-Stationary	1.51E-27	Stationary
CA_3_HOBBIES_2	0.013365331	Stationary	1.31E-23	Stationary
CA_3_HOUSEHOLD_1	0.308180045	Non-Stationary	9.19E-24	Stationary
CA_3_HOUSEHOLD_2	0.03320509	Stationary	5.34E-25	Stationary
CA_4_FOODS_1	0.026127811	Stationary	6.44E-25	Stationary
CA_4_FOODS_2	0.130139814	Non-Stationary	2.02E-30	Stationary
CA_4_FOODS_3	0.118663688	Non-Stationary	1.02E-27	Stationary
CA_4_HOBBIES_1	0.246507618	Non-Stationary	1.34E-25	Stationary
CA_4_HOBBIES_2	0.000775528	Stationary	3.57E-29	Stationary
CA_4_HOUSEHOLD_1	0.643451726	Non-Stationary	2.36E-25	Stationary
CA_4_HOUSEHOLD_2	0.526158976	Non-Stationary	5.37E-25	Stationary
TX_1_FOODS_1	0.036800071	Stationary	6.37E-27	Stationary
TX_1_FOODS_2	0.00028336	Stationary	0	Stationary

Table 4. Stationarity check Results (Augmented Dicky fuller test) (*Continued*)

TX_1_FOODS_3	0.008214364	Stationary	0	Stationary
TX_1_HOBBIES_1	0.185985526	Non-Stationary	3.90E-25	Stationary
TX_1_HOBBIES_2	0.01086493	Stationary	4.98E-22	Stationary
TX_1_HOUSEHOLD_1	0.33596135	Non-Stationary	1.93E-25	Stationary
TX_1_HOUSEHOLD_2	0.409465622	Non-Stationary	1.20E-26	Stationary
TX_2_FOODS_1	0.074771194	Non-Stationary	1.21E-27	Stationary
TX_2_FOODS_2	0.004797366	Stationary	0	Stationary
TX_2_FOODS_3	0.078298352	Non-Stationary	3.45E-30	Stationary
TX_2_HOBBIES_1	0.112566879	Non-Stationary	2.14E-26	Stationary
TX_2_HOBBIES_2	1.83E-09	Stationary	1.27E-26	Stationary
TX_2_HOUSEHOLD_1	0.159884557	Non-Stationary	1.06E-24	Stationary
TX_2_HOUSEHOLD_2	0.071650135	Non-Stationary	1.77E-22	Stationary
TX_3_FOODS_1	0.092315373	Non-Stationary	2.04E-26	Stationary
TX_3_FOODS_2	0.108362186	Non-Stationary	0	Stationary
TX_3_FOODS_3	0.02654422	Stationary	3.46E-30	Stationary
TX_3_HOBBIES_1	0.614753975	Non-Stationary	6.48E-24	Stationary
TX_3_HOBBIES_2	0.001758733	Stationary	1.37E-27	Stationary
TX_3_HOUSEHOLD_1	0.30490459	Non-Stationary	2.89E-26	Stationary
TX_3_HOUSEHOLD_2	0.515269898	Non-Stationary	6.77E-25	Stationary
WI_1_FOODS_1	0.000241426	Stationary	3.87E-25	Stationary
WI_1_FOODS_2	0.876724594	Non-Stationary	2.16E-30	Stationary
WI_1_FOODS_3	0.582192651	Non-Stationary	1.68E-27	Stationary
WI_1_HOBBIES_1	0.016838784	Stationary	6.35E-27	Stationary
WI_1_HOBBIES_2	0.006290415	Stationary	7.04E-21	Stationary
WI_1_HOUSEHOLD_1	0.231552217	Non-Stationary	4.47E-26	Stationary
WI_1_HOUSEHOLD_2	0.172272631	Non-Stationary	1.19E-24	Stationary
WI_2_FOODS_1	0.114706355	Non-Stationary	3.82E-30	Stationary
WI_2_FOODS_2	0.907627417	Non-Stationary	0	Stationary
WI_2_FOODS_3	0.565704064	Non-Stationary	0	Stationary
WI_2_HOBBIES_1	0.388960834	Non-Stationary	7.00E-27	Stationary
WI_2_HOBBIES_2	0.001218291	Stationary	4.29E-28	Stationary
WI_2_HOUSEHOLD_1	0.338079811	Non-Stationary	0	Stationary
WI_2_HOUSEHOLD_2	0.390617085	Non-Stationary	1.27E-27	Stationary
WI_3_FOODS_1	0.088189069	Non-Stationary	2.47E-30	Stationary
WI_3_FOODS_2	0.283054378	Non-Stationary	0	Stationary

Table 5. Stationarity check Results (Augmented Dicky fuller test) (*Continued*)

WI_3_FOODS_3	0.168658341	Non-Stationary	0	Stationary
WI_3_HOBBIES_1	1.68E-05	Stationary	1.53E-27	Stationary
WI_3_HOBBIES_2	0.000275599	Stationary	2.89E-25	Stationary
WI_3_HOUSEHOLD_1	0.055660951	Non-Stationary	1.27E-29	Stationary
WI_3_HOUSEHOLD_2	0.134125657	Non-Stationary	5.35E-24	Stationary

The basic translation of a non-stationary time series into a stationary time series can be accomplished by taking the differences in time series. The first-order difference is typically sufficient for non-seasonal data to achieve apparent stationarity. The differenced time series can be taken by using the equation shown below.

$$\hat{y}_t = y_t - y_{t-1} \tag{3}$$

3.4 Model Evaluation matrices

The challenge of calculating the accuracy of the forecast is related to the need to select an acceptable error measure. In particular, choosing an error measure to determine the precision of forecasts over time series is an essential subject for forecasting research (Davydenko & Fildes, 2013). Model evaluation was conducted using two metrics.

- Mean absolute error (MAE)
- Mean absolute percentage error (MAPE)

The mean absolute error (MAE) value is calculated in the same units as the data. Since it does not square the errors in the equation, it is less susceptible to the occasional substantial error.

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|A_t - F_t| \tag{4}$$

Mean absolute percentage error (MAPE) is widely used in evaluating time series forecasting. The MAPE value of each time series was taken to assess the GRU model. Since the MAPE value gives the percentage value, it is easier to interpret the error.

$$MAPE = \frac{1}{n}\left(\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right|\right) * 100 \tag{5}$$

4. Results

4.1 Hyperparameter Tuning in GRU

Hyperparameters were tuned using the BayesSearchcv class in the sci-kit learn library. After fitting the Bayes search object with the training data, the result of the object is shown in 4. The Table shows the hyperparameter values used in each iteration. When training the model across a 5-fold time series, cross-validation techniques were used to ensure that the model was not overfitted. The mean test score is calculated using the negative mean value for MSE values for 5 test splits. The Std train test score column in Table 2 is calculated by taking the negative mean value for MSE values for five train splits. Also, the mean train score is shown in Table 6.

According to the Bayes Search hyperparameter, the best model was selected using the mean score test value after tuning run through 25 iterations. The best model was chosen by getting the model with the least MSE for the testing sets. This can also be seen in the mean test score, the negative MSE for the testing sets. According to the test score with its hyperparameters values, the best model is the last row of Table 6. The graph in
Figure **6** shows how the MSE changed with each epoch for the training data. MSE has drastically reduced during the first 10 epochs, then after the 60th epoch, the error becomes more stable.

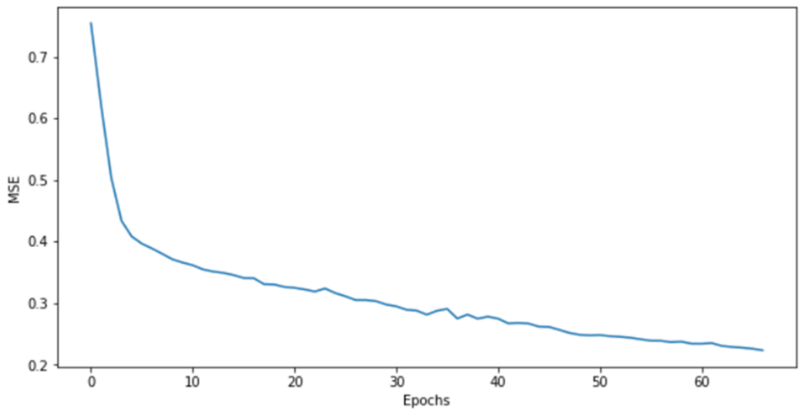


Figure 6. MSE change over the epochs

4.2 LightGBM model

LightGBM model Hyperparameters were also tuned using BayesSearchcv. Table 5 shows the hyperparameter values used in each iteration and train test scores.

Table 6. Bayes search optimization results for GRU model

Mean test score	*std test score*	*rank test score*	*mean train score*	*std train score*	*activation*	*batch size*	*dropout*	*epochs*	*layer 1 units*	*layer 2 units*	*layer 3 units*	*rank train score*
-0.9345	0.1577	22	-0.5747	0.0607	relu	254	0.7430	91	305	399	136	23
-0.7011	0.1694	15	-0.2388	0.0081	relu	42	0.2860	40	164	313	100	11
-0.8036	0.3608	20	-0.2071	0.0054	relu	219	0.1160	65	88	301	240	6
-0.6538	0.1324	9	-0.1937	0.0062	relu	49	0.1809	51	373	88	246	5
-0.6981	0.1694	13	-0.2535	0.0157	relu	225	0.3143	68	240	117	276	13
-1.0224	0.2705	25	-0.7007	0.0422	relu	118	0.7999	19	229	255	119	24
-0.7339	0.0580	18	-0.4503	0.0502	tanh	134	0.2873	6	73	56	249	20
-0.7504	0.1093	19	-0.3635	0.0355	relu	31	0.7138	99	148	355	251	19
-0.8854	0.1516	21	-0.5074	0.0424	relu	200	0.2752	9	322	246	99	22
-0.6851	0.0972	12	-0.3060	0.0304	relu	94	0.4484	62	58	155	160	18
-0.7122	0.1179	17	-0.1482	0.0414	tanh	2	0.3662	100	50	400	400	2
-0.7066	0.1712	16	-0.2120	0.0078	relu	41	0.3162	71	200	65	224	8
-0.6674	0.1218	11	-0.2991	0.0095	tanh	215	0.6115	58	354	50	400	17
-1.0216	0.2209	24	-0.7204	0.1359	relu	226	0.7928	43	241	50	252	25
-0.5858	0.0867	4	-0.2387	0.0075	tanh	253	0.4241	98	214	323	116	10
-0.5893	0.0738	5	-0.1800	0.0090	tanh	256	0.2360	100	329	272	185	4
-0.5810	0.0710	3	-0.2087	0.0067	tanh	256	0.3366	100	284	295	158	7
-0.5795	0.0825	2	-0.2673	0.0125	tanh	256	0.4515	100	279	253	50	14

Table 7. Bayes search optimization results for GRU model (*Continued*)

-0.9746	0.2696	23	-0.4968	0.0099	tanh	256	0.8000	100	400	50	50	21
-0.7006	0.0993	14	-0.1115	0.0426	tanh	2	0.1000	100	400	50	113	1
-0.6555	0.1214	10	-0.2373	0.0076	tanh	215	0.4926	100	400	50	400	9
-0.6307	0.1034	8	-0.2728	0.0112	tanh	256	0.3057	100	109	50	50	15
-0.5994	0.0852	6	-0.2771	0.0143	tanh	256	0.4832	100	182	361	50	16
-0.6089	0.0892	7	-0.1621	0.0083	tanh	256	0.2703	100	400	358	400	3
-0.5405	0.0739	1	-0.2139	0.0026	tanh	128	0.1980	67	300	300	300	12

Table 8. Bayes search optimization results for the LightGBM model

mean_test _score	*std_test_s core*	*param_max _depth*	*param_min_ data_in_leaf*	*param_n_ estimators*	*param_num _leaves*	*rank_test _score*	*mean_train _score*	*std_train _score*	*rank_train _score*
0.9585	0.0091	66	271	118	484	16	0.9803	0.0014	22
0.9593	0.0093	397	169	303	81	6	0.9877	0.0013	7
0.9565	0.0100	498	456	357	292	24	0.9818	0.0015	18
0.9598	0.0097	375	69	323	156	3	0.9948	0.0018	4
0.9591	0.0102	483	54	386	251	9	0.9977	0.0013	1
0.9585	0.0094	339	292	403	161	15	0.9866	0.0010	10
0.9588	0.0095	383	241	329	251	11	0.9873	0.0010	9
0.9587	0.0093	144	320	446	58	13	0.9849	0.0011	12
0.9591	0.0100	66	73	448	345	8	0.9969	0.0010	2
0.9572	0.0095	457	451	373	377	20	0.9822	0.0013	17
0.9566	0.0098	436	500	464	53	23	0.9816	0.0014	19
0.9566	0.0100	56	499	468	191	22	0.9822	0.0015	15
0.9601	0.0097	336	59	160	186	1	0.9922	0.0019	6
0.9556	0.0102	280	499	58	377	25	0.9745	0.0020	25
0.9597	0.0099	89	59	52	378	4	0.9864	0.0012	11
0.9574	0.0093	201	321	51	55	18	0.9756	0.0017	24
0.9593	0.0094	309	186	99	290	7	0.9822	0.0012	16
0.9569	0.0095	493	454	96	335	21	0.9766	0.0017	23
0.9595	0.0100	440	50	50	487	5	0.9876	0.0011	8
0.9574	0.0095	282	431	306	442	19	0.9816	0.0013	20
0.9600	0.0098	57	50	50	128	2	0.9839	0.0020	13
0.9586	0.0093	64	295	171	481	14	0.9815	0.0013	21
0.9584	0.0099	396	174	496	491	17	0.9933	0.0006	5
0.9589	0.0094	326	268	208	304	10	0.9834	0.0012	14
0.9588	0.0101	390	127	445	309	12	0.9951	0.0004	3

4.3 VAR model

Selecting lag length is essential in VAR modelling. Error, trend, and seasonal models and ARIMA models lead to choosing the best by minimizing various information criteria like the Akaike Information Criterion (AIC) (an in-sample match technique to measure a model's probability of estimating future values) (Akaike, 1974), which are usually described by minimizing a one-step square error in the forecast (Makridakis et al., 2020). Several information criteria can be used to obtain the best lag value. AIC can be used to find the best lag value; a good model has minimum AIC among all the different lag value models. Another criterion for selecting the best lag length that tests the trade-off between model fit and model complexity is the Bayesian information criterion (BIC) (Stone, 1979). A better fit is shown by a lower BIC score, just like the AIC score. The Final Prediction Error (FPE) was also used to estimate the model-fitting error for different lag lengths. Similar to the other 2 criteria, the model that gives the minimum FPE selects the best model.

Table 9. Lag value selection criterion

Lag length	*AIC*	*BIC*	*FPE*	*HQIC*
0	555.1	555.3	1.17E+241	555.2
1	540	554.5	3.24E+234	545.3
2	535	563.9	2.31E+232	545.6
3	533.1	576.2	3.45E+231	549
4	532.1	589.6	1.41E+231	553.3
5	531.3	603.2	7.70E+230	557.8
6	531.1	617.2	7.44E+230	562.8
7	531.7	632.2	1.94E+231	568.7
8	532.5	647.3	6.67E+231	574.8
9	533.2	662.3	2.22E+232	580.7
10	533.7	677.1	7.57E+232	586.5
11	534	691.8	2.64E+233	592.1
12	533.9	706	7.50E+233	597.3

Table 9 shows the information criterion values for different lag lengths. The tables also show the Hannan-Quinn information criterion (HQIC) values, which is also a measure of the goodness of fit of a statistical model. HQIC is similar to AIC but not based on the log-likelihood function. According to the table, lag lengths 1 and 6 are the lowest when considering the 4 criteria represented in the columns. The lag length was chosen as 6 since using just 1 previous time step is too small to capture the data patterns.

4.4 Model Comparison

Using GRU, LightGBM, and VAR models' predictions were done 30 days ahead, considering each department in each store as a multivariate time series and the forecasts of the 7 departments of CA_1 using both models are shown in Figure 7. Predictions of the other stores are provided in the appendix.

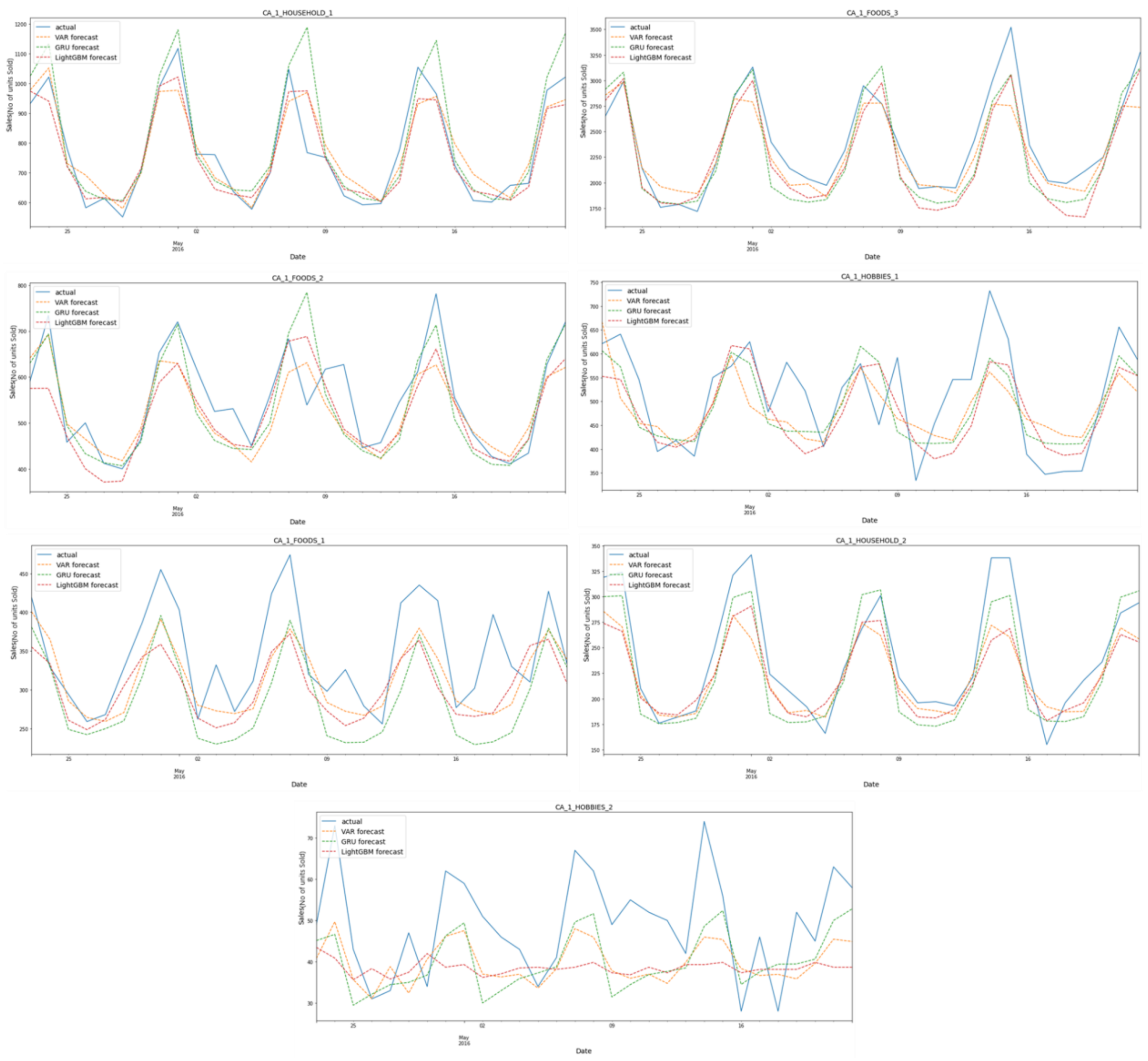


Figure 7. Predictions vs Actual plots of store CA_1

GRU, LightGBM, and VAR models were compared using MAE and MAPE. The mean MAPE value for predictions of the GRU, VAR, and LightGBM models were 13.77%, 14.87%, and 14.24%, respectively.

The mean MAE value of the GRU model is 68 units, and 72 units for the VAR model. The standard deviation of the GRU model MAE is 58 units, which is lower than the VAR model's standard deviation of 66 units. MAPE values for the GRU and VAR models in each of the 7 departments (series) that the SKUs populate on the retail chain were also reviewed. In the department, FOOD_1 FOOD_2, HOBBIES_1, HOBBIES_2, and HOUSEHOLD_2 GRU model has comparatively lower MAPE values. VAR model's MAPE values are less in FOODS_3 and HOUSEHOLD_1 departments. According to the MAE values, the GRU model has performed The mean MAE value of the GRU model is 68 units, and 72 units for the VAR model. The standard deviation of the GRU model MAE is 58 units, which is lower than the VAR model's standard deviation of 66 units. MAPE values for the GRU and VAR models in each of the 7 departments (series) that the SKUs populate on the retail chain were also reviewed. In the department, FOOD_1 FOOD_2, HOBBIES_1, HOBBIES_2, and HOUSEHOLD_2 GRU model has comparatively lower MAPE values. VAR model's MAPE values are less in FOODS_3 and HOUSEHOLD_1 departments. According to the MAE values, the GRU model has performed better in 5 departments and the VAR model in 2 departments. Therefore, considering the MAPE values, the GRU model performed better than the VAR model. Despite this, the LightGBM model performed better in 4 departments.

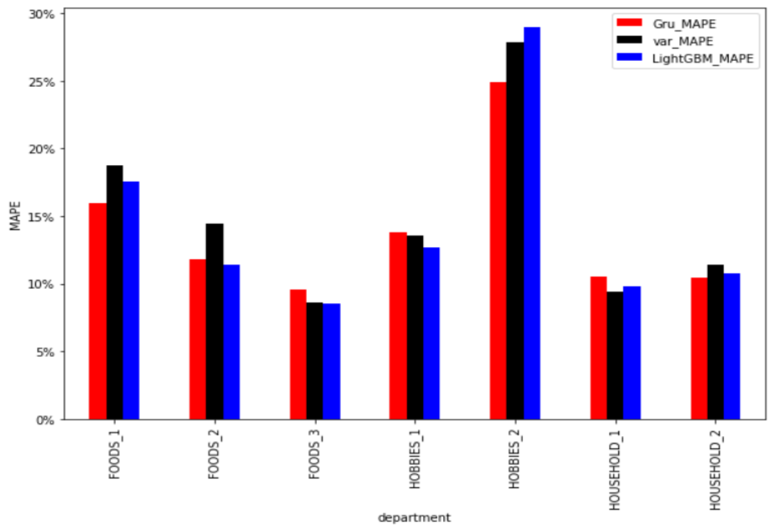


Figure 8. Model MAPE values of departments

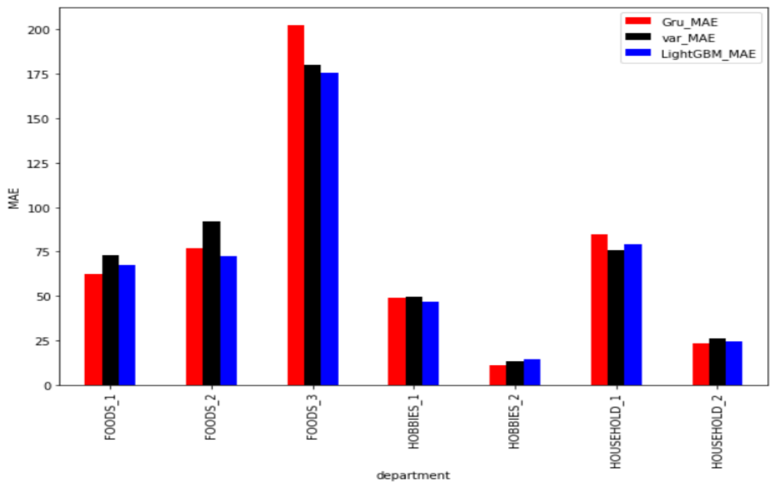


Figure 9. Models MAE values of departments

Managerial Insights

To increase demand forecasting accuracy, organizations should invest in creating and implementing sophisticated forecasting models such as Artificial Neural Networks (ANNs) and Gated Recurrent Units (GRUs). Organizations could use multivariate time series forecasting techniques when numerous variables are interrelated to effectively manage correlations and complexity between variables and produce educated forecasts. Consider employing hybrid forecasting models, which combine the advantages of classic statistical approaches with the capabilities of deep learning techniques.

Conclusion

The goal of this paper is to find where GRU can be used to forecast the demand for multiple stores. To achieve the expected goal of the research, 3 objectives were defined. The first objective is to build the GRU model to predict the demand for the data collected from the M5 forecasting challenge. The second objective is to create a VAR model for the same dataset. The third and final objective is to compare the 2 models.

To achieve the first objective, the GRU model was built using Python. All SKU sales data were grouped by department so the model could use each store's department as input during the modelling process. Hyperparameters of the model were tuned using Bayesian optimization techniques. Bayesian hyperparameter search was run through 25 iterations to find a good forecasting model.

The second objective of the research is to build VAR and LightGBM models to predict the sales for 30 days for the same 70-time series from each store's departments. First-order differencing was used to make the time series stationary before VAR modelling after the stationarity check lag length of the VAR model was chosen as 6 using the AIC and BIC criterion.

The third objective of the research is to evaluate and compare the GRU, LightGBM, and VAR models based on their predictions, which were done for 30 days. Both models were evaluated using MAE and MAPE values generated for the forecasts of 7 departments in the 10 stores, which consists of 70 different time-series predictions. The MAPE value for the GRU model was 13.77%, and the VAR model was 14.87%. LightGBM GBM model MAPE value is 14.24%. It was seen that the GRU model's overall performance was better than the VAR model and LightGBM model. The mean MAE value of the GRU model was 68 units, and for the VAR model, 72 units. The mean MAE value of the LightGBM model was 69 units. The GRU model MAE standard deviation was 58 units, lower than the VAR model standard deviation of 66 units. The standard deviation of the LightGBM model was 58. From the research findings, we can conclude that the GRU neural network approach to forecasting the demand for multiple stores is more effective when compared to VAR for the subject dataset.

This research studies how the GRU model can be used to forecast the sales of multiple stores. In this research, only the previous sales data from each store were used as inputs to the model. However, it might be interesting to investigate other variables that can be used to improve the model, such as a special event (Mother's Day, Father's Day, Christmas, Easter, etc.). While a Sales Forecasting System Based On Deep Learning Techniques presents a novel method of sales forecasting via Deep neural networks, evaluating the larger context of sales forecasting is critical. We can extend the model to forecast the demand during post-promotional periods(Hewage et al., 2022). Another possible future implementation of this research is to develop the GRU model to predict sales during a pandemic. Demand may change drastically during a pandemic because customers' purchasing habits change to mitigate supply uncertainty. Modelling customer demand using disease proliferation and other auxiliary data, such as government decisions to sudden lockdown days, are possible areas of extension in this regard.

Further, investigate and create innovative hybrid forecasting models that successfully blend old statistical approaches with new deep learning techniques. Investigate various combinations and architectures of these models to improve accuracy and interpretability in real-world forecasting scenarios. Improve uncertainty estimation in forecasting models. Develop approaches for quantifying and communicating the uncertainty of various forecasting scenarios, allowing decision-makers to make better-informed choices.

Deep neural networks, for example, can have lengthy training cycles and demand significant computer resources. This can be a problem, especially when dealing with massive datasets. When using forecasting models in real-world corporate contexts, obstacles such as resistance to change, data integration concerns, and the need for continuing model maintenance may arise.

Acknowledgments

The authors wish to acknowledge the support of grant ID SRC/LT/2020/20 provided by the Senate Research Committee of the University of Moratuwa for its support in making this research possible.

References

Azubuike, I., & Kosemoni, O. (2017). A Comparison of Univariate and Multivariate Time Series Approaches to Modeling Currency Exchange Rate. *British Journal of Mathematics & Computer Science*, *21*(4), 1–17. https://doi.org/10.9734/bjmcs/2017/30733

Ahaggach, H., Abrouk, L., & Lebon, E. (2024). Systematic Mapping Study of Sales Forecasting: Methods, Trends, and Future Directions. *Forecasting, 6*(3), 502-532. https://doi.org/10.3390/FORECAST6030028

Arachchige, A., Sugathadasa, R., Herath, O., & Thibbotuwawa, A. (2021). Artificial Neural Network Based Demand Forecasting Integrated With Federal Funds Rate. Applied Computer Science, 17(4), 34–44. https://doi.org/10.23743/acs-2021-27

Bandara, K., Shi, P., Bergmeir, C., Hewamalage, H., Tran, Q., & Seaman, B. (2019). Sales Demand Forecast in E-commerce using a Long Short-Term Memory Neural Network Methodology. *Springer International Publishing*. http://arxiv.org/abs/1901.04028

Chang, Y.-Y., Sun, F.-Y., Wu, Y.-H., & Lin, S.-D. (2018). A Memory-Network Based Solution for Multivariate Time-Series Forecasting. *ArXiv*. http://arxiv.org/abs/1809.02105

Chen, Z., Ma, M., Li, T., Wang, H., & Li, C. (2023). Long sequence time-series forecasting with deep learning: A survey. *Information Fusion*, *97*, 101819. https://doi.org/10.1016/J.INFFUS.2023.101819

Cochran, J. J., Cox, L. A., Keskinocak, P., Kharoufeh, J. P., Smith, J. C., Wang, S., & Chaovalitwongse, W. A. (2011). Evaluating and Comparing Forecasting Models. In *Wiley Encyclopedia of Operations Research and Management Science*. John Wiley & Sons, Inc. https://doi.org/10.1002/9780470400531.eorms0307

Crone, S. F., Hibon, M., & Nikolopoulos, K. (2011). Advances in forecasting with neural networks? Empirical evidence from the NN3 competition on time series prediction. *International Journal of Forecasting*, *27*(3), 635–660. https://doi.org/10.1016/j.ijforecast.2011.04.001

Davydenko, A., & Fildes, R. (2013). Measuring Forecasting Accuracy: The Case Of Judgmental Adjustments To Sku-Level Demand Forecasts. *International Journal of Forecasting*, *29*(3), 510–522. https://doi.org/10.1016/j.ijforecast.2012.09.002

Eglite, L., & Birzniece, I. (2022). Retail Sales Forecasting Using Deep Learning: Systematic Literature Review. *Complex Systems Informatics and Modeling Quarterly*, *2022*(30). https://doi.org/10.7250/csimq.2022-30.03

Forslund, H., & Jonsson, P. (2007). The impact of forecast information quality on supply chain performance. *International Journal of Operations and Production Management*, *27*(1), 90–107. https://doi.org/10.1108/01443570710714556

Gilbert, K. (2005). An ARIMA supply chain model. In *Management Science* (Vol. 51, Issue 2, pp. 305–310). https://doi.org/10.1287/mnsc.1040.0308

He, Q. Q., Wu, C., & Si, Y. W. (2022). LSTM with particle Swam optimization for sales forecasting. *Electronic Commerce Research and Applications*, *51*. https://doi.org/10.1016/j.elerap.2022.101118

Helmini, S., Jihan, N., Jayasinghe, M., & Perera, S. (2019). *Sales forecasting using multivariate long short term memory network models*. https://doi.org/10.7287/peerj.preprints.27712v1

Hewage, H. C., & Perera, H. N. (2021). Comparing Statistical and Machine Learning Methods for Sales Forecasting during the Post-promotional Period. *2021 IEEE International Conference on Industrial Engineering and Engineering Management, IEEM 2021*. https://doi.org/10.1109/IEEM50564.2021.9672954

Hewage, H. C., Perera, H. N., & De Baets, S. (2022). Forecast adjustments during post-promotional periods. *European Journal of Operational Research*, *300*(2). https://doi.org/10.1016/j.ejor.2021.07.057

Hewamalage, H., Bergmeir, C., & Bandara, K. (2020). Recurrent Neural Networks for Time Series Forecasting: Current status and future directions. *International Journal of Forecasting*. https://doi.org/10.1016/j.ijforecast.2020.06.008

Hyndman, R. J. (2020). A brief history of forecasting competitions. *International Journal of Forecasting*, *36*(1), 7–14. https://doi.org/10.1016/j.ijforecast.2019.03.015

Kolassa, S. (2016). Evaluating predictive count data distributions in retail sales forecasting. *International Journal of Forecasting*, *32*(3), 788–803. https://doi.org/10.1016/j.ijforecast.2015.12.004

Lai, K. K. (2006). An Integrated Data Preparation Scheme for Neural Network Data Analysis. *IEEE Transactions on Knowledge and Data Engineering*, *18*(2), 217–230. https://doi.org/10.1109/TKDE.2006.22

Li, Z., Han, J., & Song, Y. (2020). On the forecasting of high-frequency financial time series based on ARIMA model improved by deep learning. *Journal of Forecasting*, *39*(7), 1081–1097. https://doi.org/10.1002/for.2677

Lipton, Z. C., Berkowitz, J., & Elkan, C. (2015). *A Critical Review of Recurrent Neural Networks for Sequence Learning*. http://arxiv.org/abs/1506.00019

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2020). The M4 Competition: 100,000 time series and 61 forecasting methods. *International Journal of Forecasting*, *36*, 54–74. https://doi.org/10.1016/j.ijforecast.2019.04.014

Obaidur Rahman, M., Sabir Hossain, M., Shafiul Alam Forhad, M., Kamal Hossen, M., & Junaid, T.-S. (2019). Predicting Prices of Stock Market using Gated Recurrent Units (GRUs) Neural Networks. In *IJCSNS International Journal of Computer Science and Network Security* (Vol. 19, Issue 1). https://www.researchgate.net/publication/331385031

Predić, B., Jovanovic, L., Simic, V., Bacanin, N., Zivkovic, M., Spalevic, P., Budimirovic, N., & Dobrojevic, M. (2024). Cloud-load forecasting via decomposition-aided attention recurrent neural network tuned by modified particle swarm optimization. *Complex and Intelligent Systems*, *10*(2), 2249–2269. https://doi.org/10.1007/S40747-023-01265-3/FIGURES/11

Perera, H. N., Hurley, J., Fahimnia, B., & Reisi, M. (2019). The human factor in supply chain forecasting: A systematic review. *European Journal of Operational Research*, *274*(2), 574–600. https://doi.org/10.1016/j.ejor.2018.10.028

Qin, Z., Yang, S., & Zhong, Y. (2024). Hierarchically Gated Recurrent Neural Network for Sequence Modeling. *Retrieved August 19*, from https://github.com/OpenNLPLab/HGRN

Šestanović, T., & Arnerić, J. (2021). Neural network structure identification in inflation forecasting. *Journal of Forecasting*, *40*(1), 62–79. https://doi.org/10.1002/for.2698

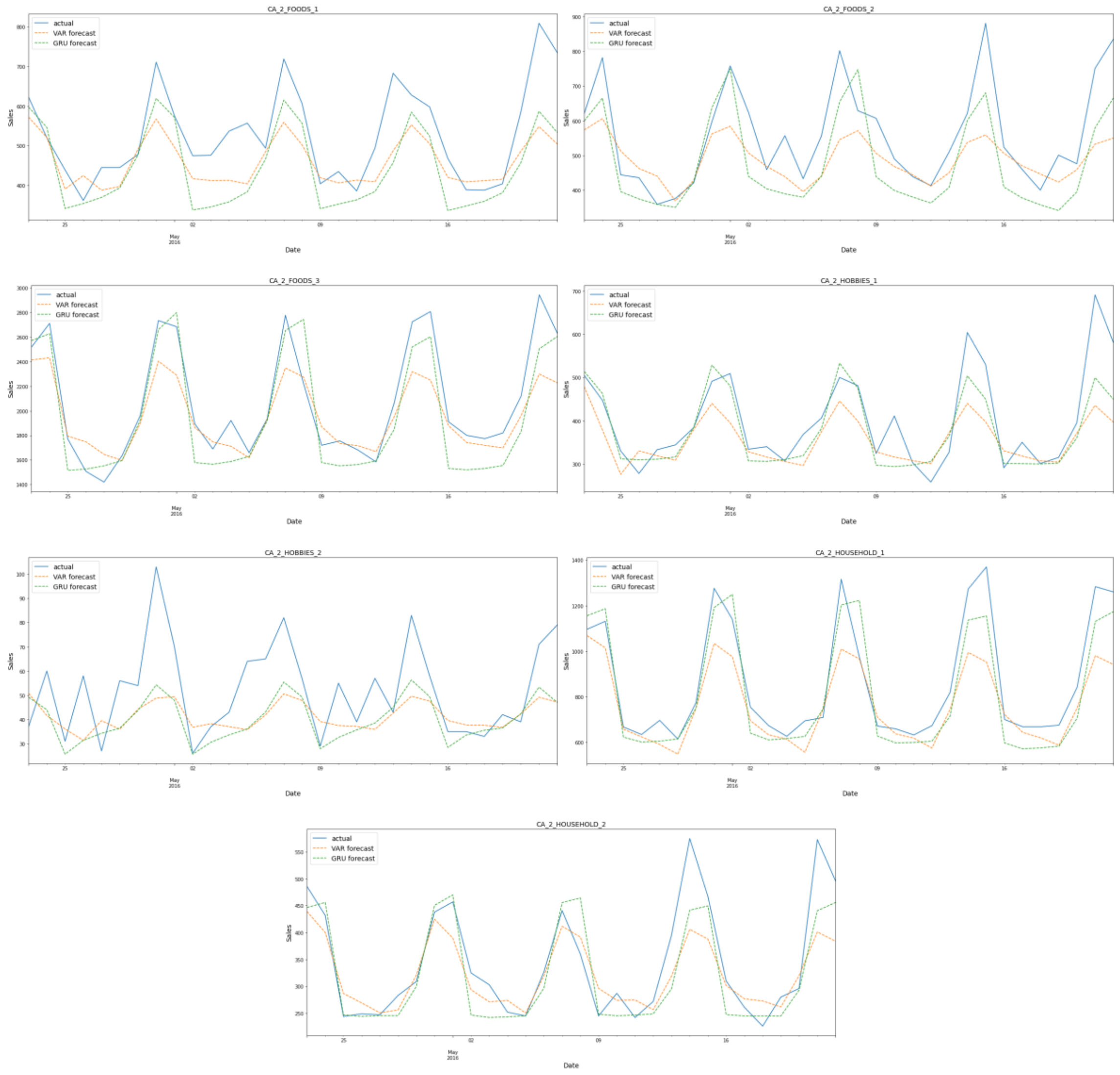
Wang, P., Gurmani, S. H., Tao, Z., Liu, J., & Chen, H. (2024). Interval time series forecasting: A systematic literature review. *Journal of Forecasting*, *43*(2), 249–285. https://doi.org/10.1002/FOR.3024

Yang, K., & Shahabi, C. (2005). On the stationarity of multivariate time series for correlation-based data analysis. *Proceedings - IEEE International Conference on Data Mining, ICDM*, 805–808. https://doi.org/10.1109/ICDM.2005.109

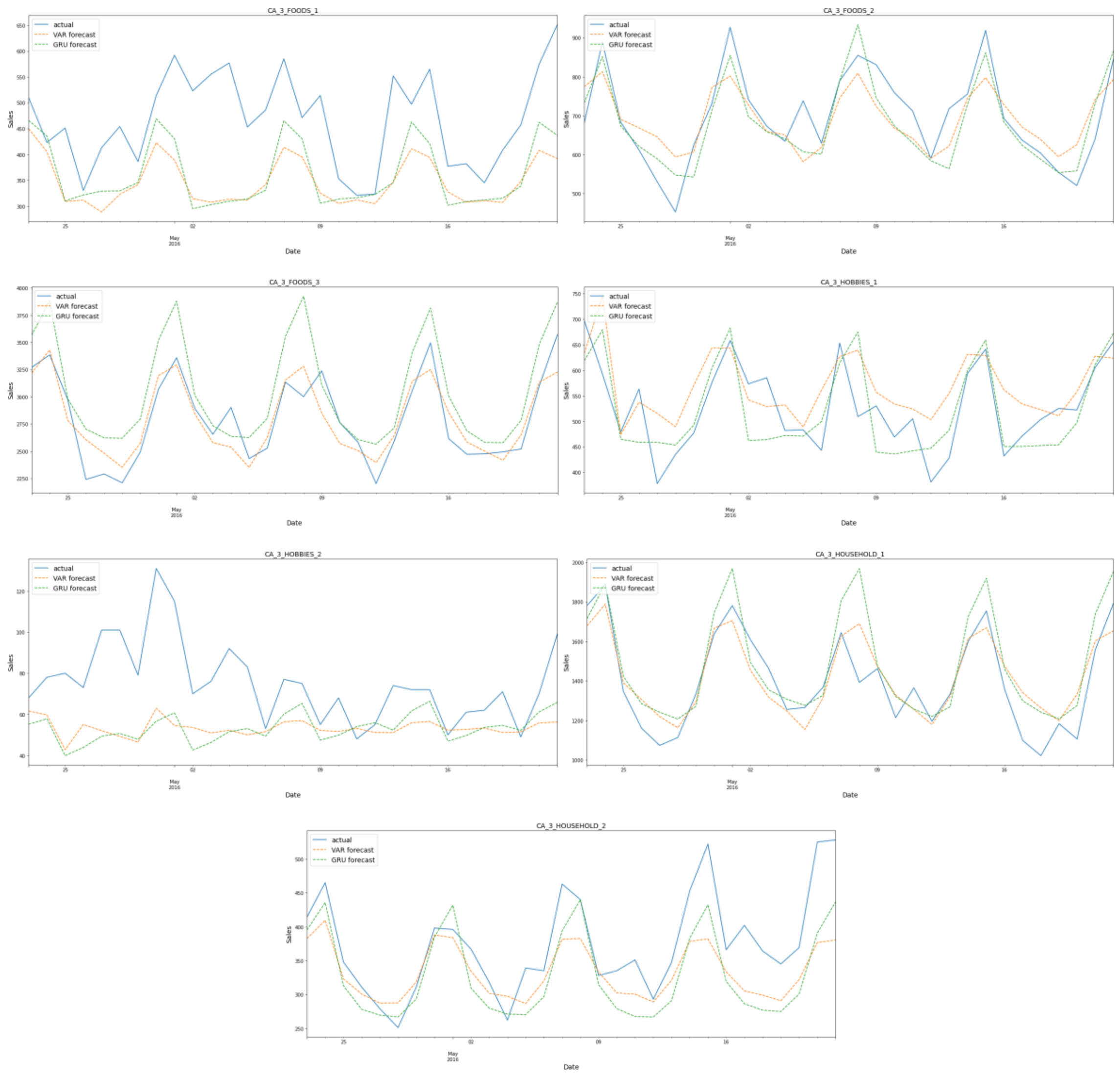
Zhang, G., Eddy Patuwo, B., & Y. Hu, M. (1998). Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting*, *14*(1), 35–62. https://doi.org/10.1016/S0169-2070(97)00044-7

Appendix A.

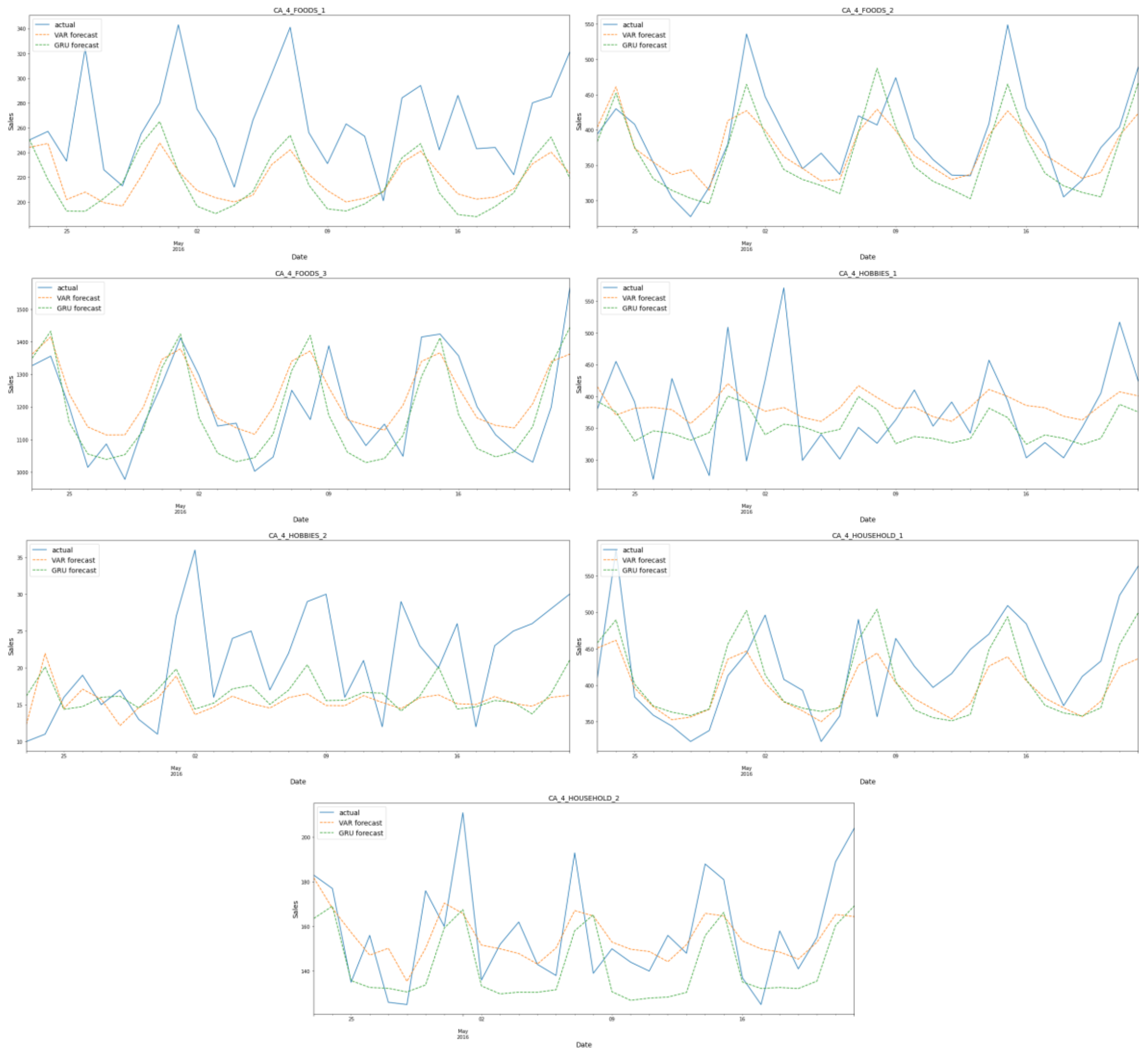
Sales forecasting plots of GRU and VAR models



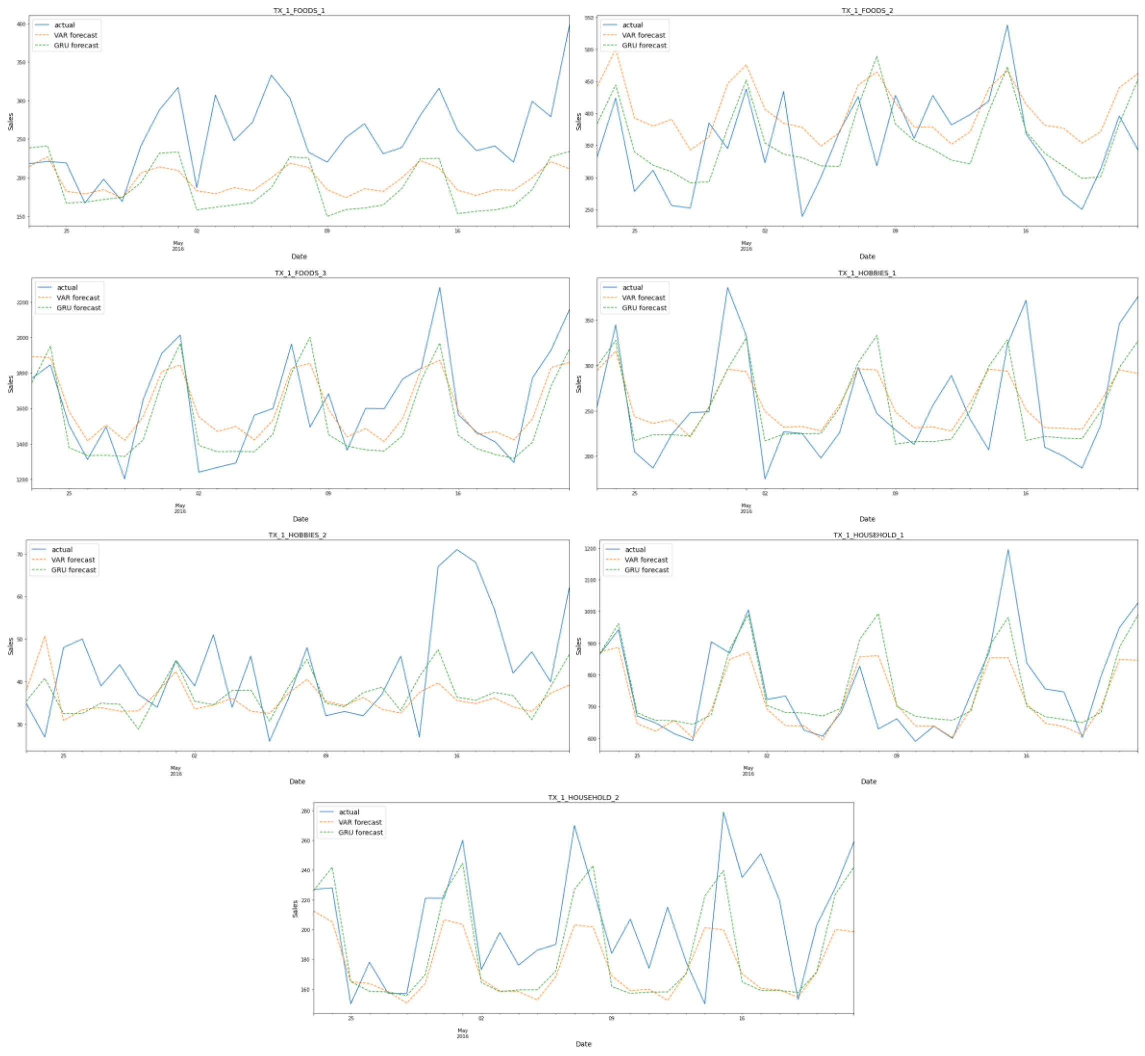
A 1:Predictions vs Actual plots of store CA_2



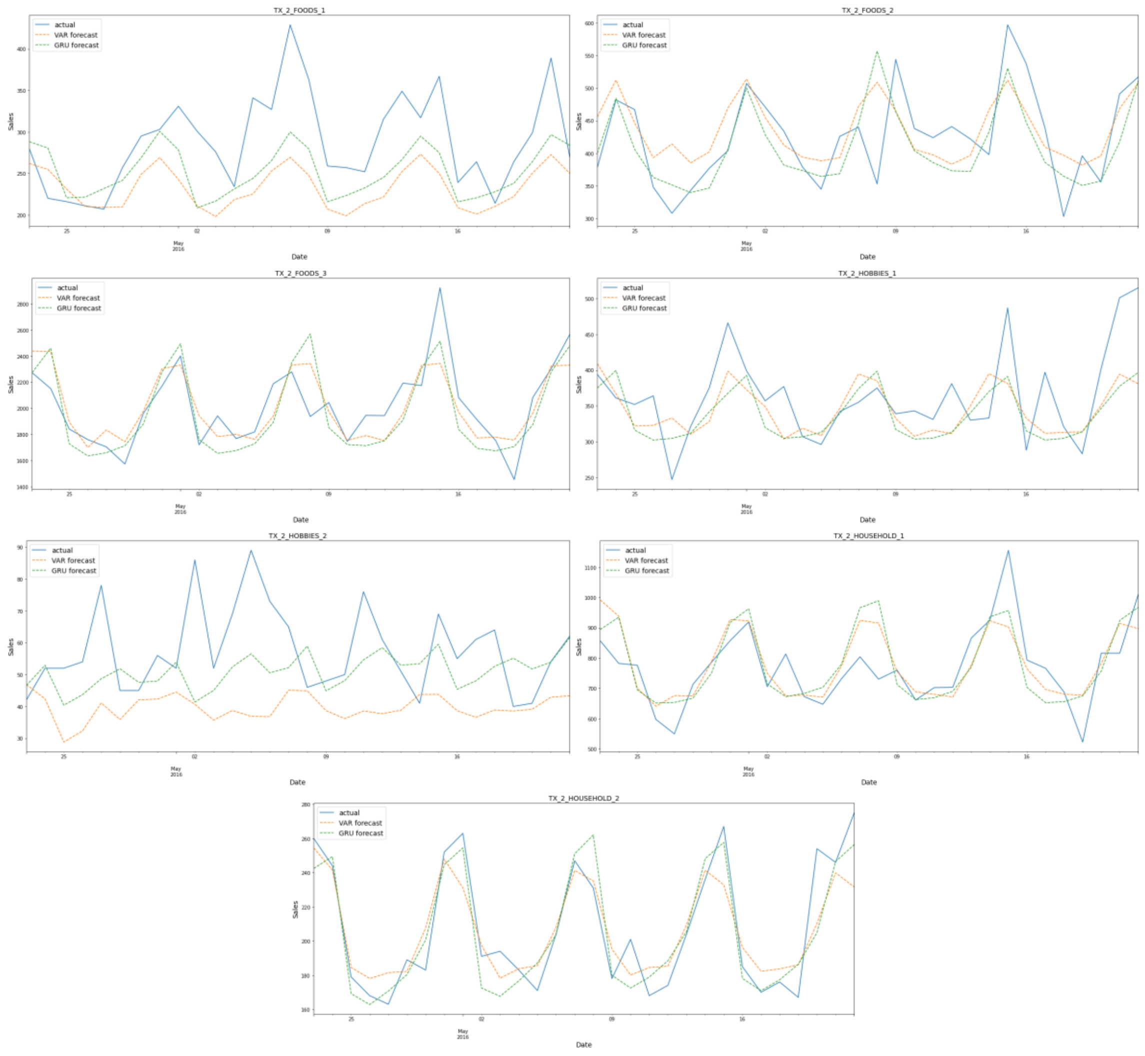
A 2: Predictions vs Actual plots of store CA_3



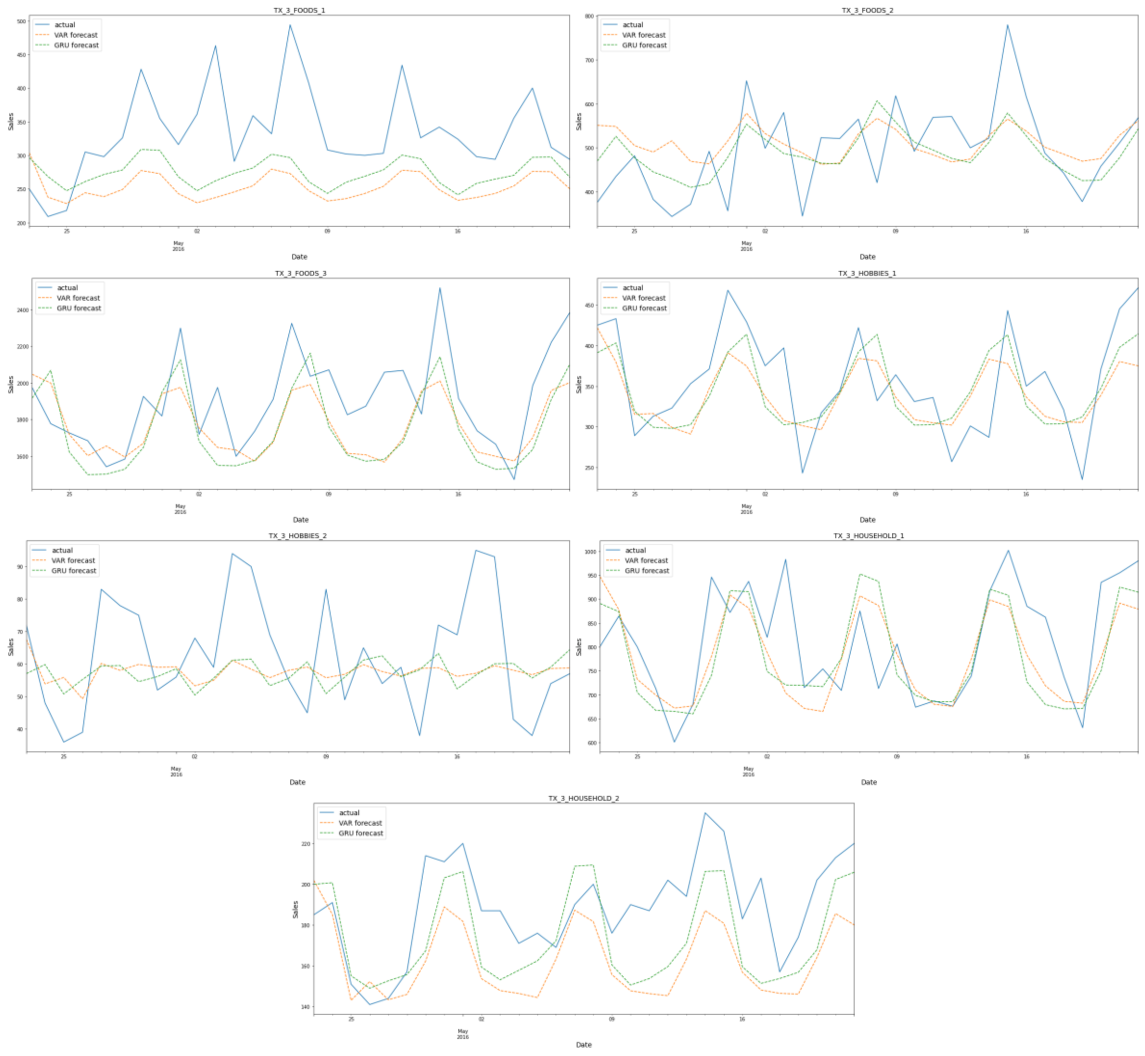
A 3: Predictions vs Actual plots of store CA_4



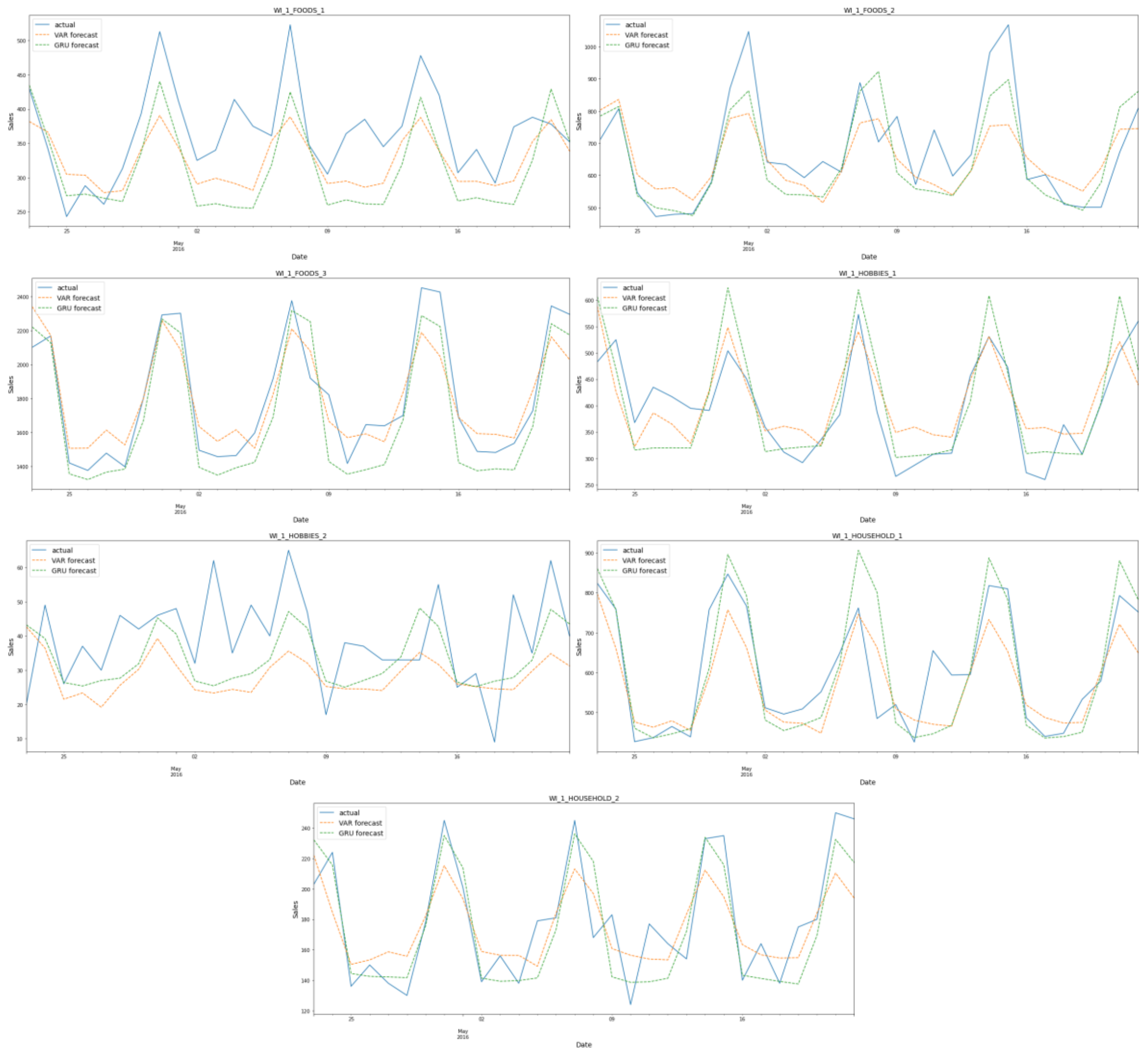
A 4: Predictions vs Actual plots of store TX_1



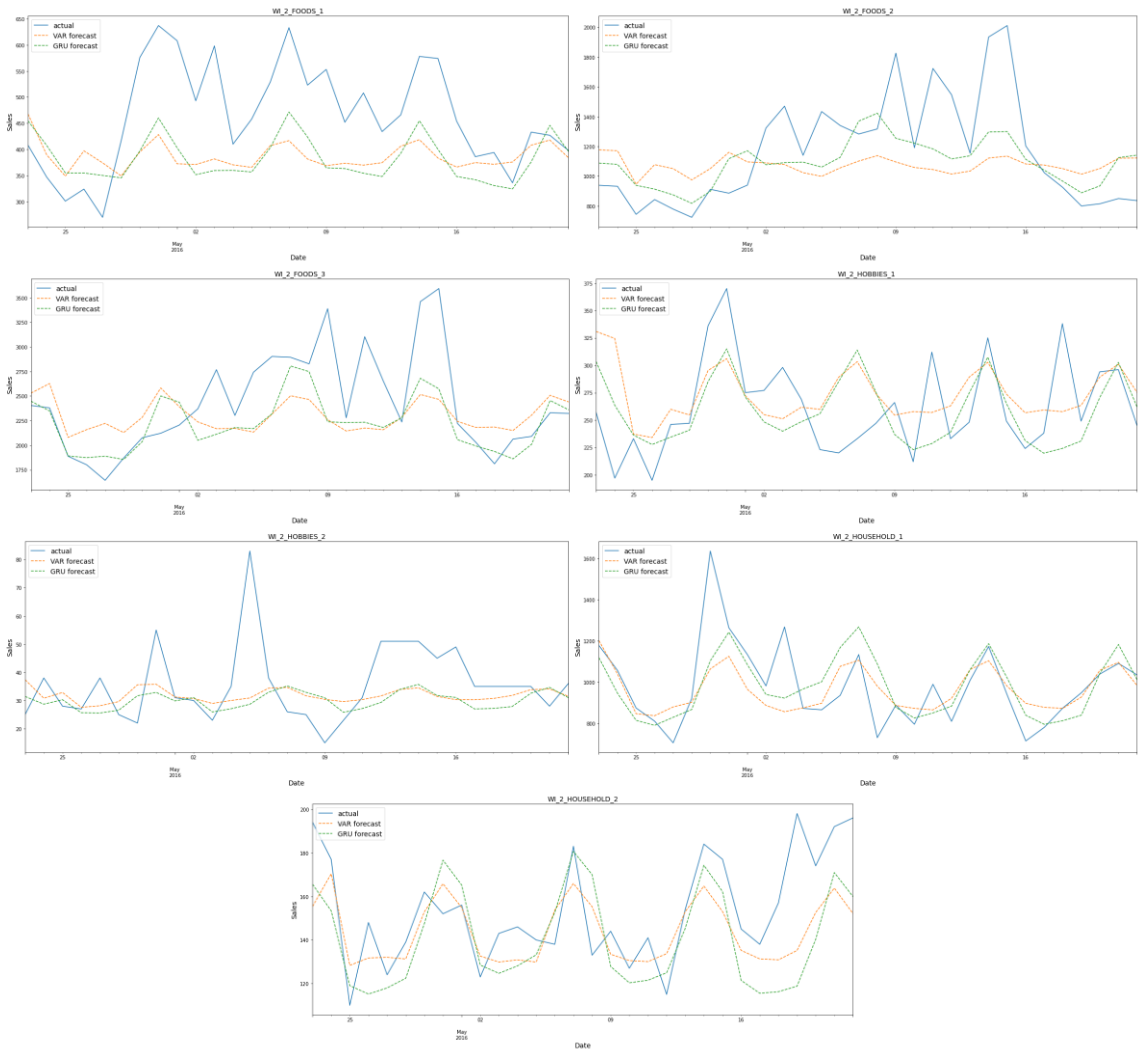
A 5: Predictions vs Actual plots of store TX_2



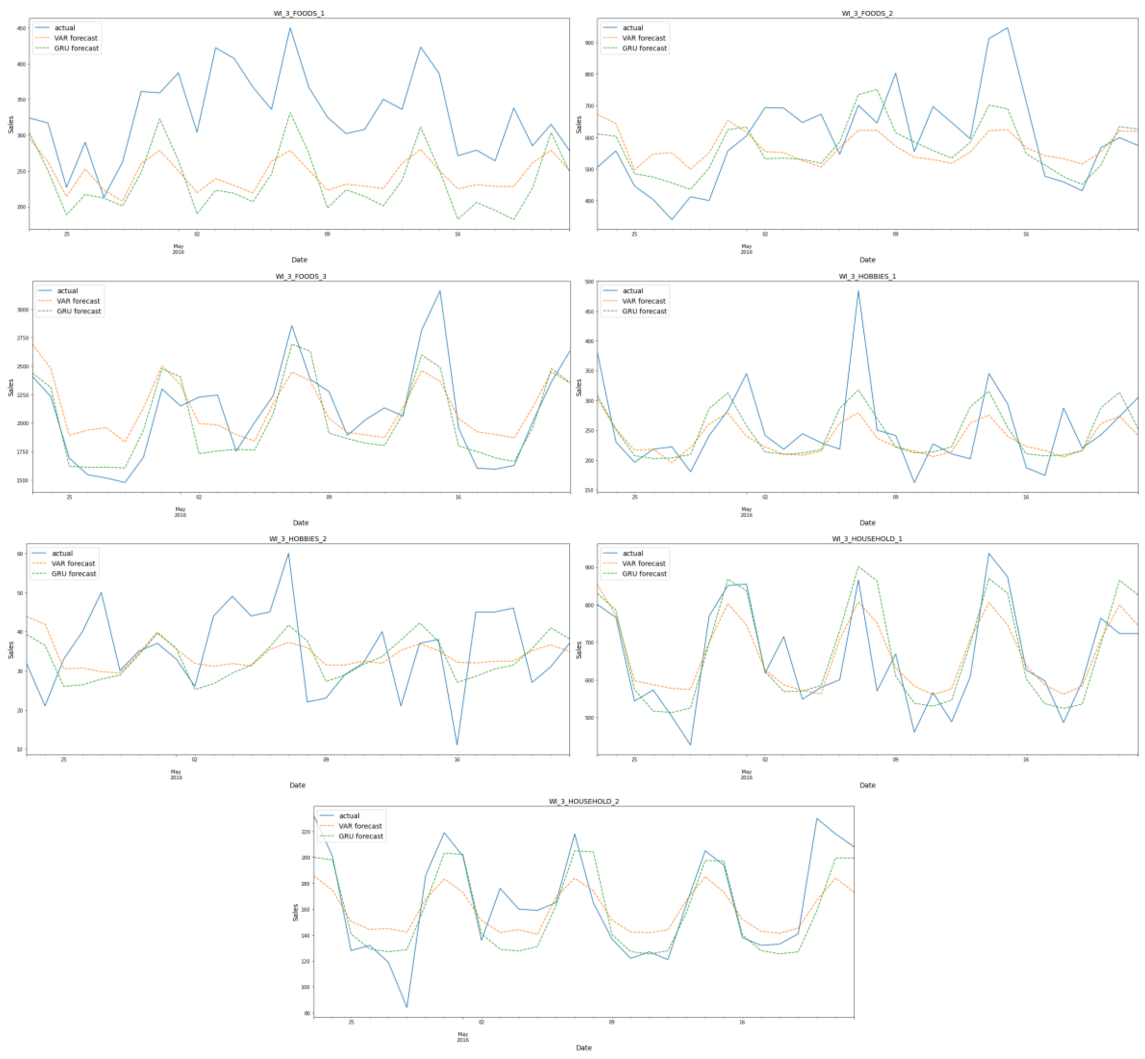
A 6: Predictions vs Actual plots of store TX_3



A 7: Predictions vs Actual plots of store WI_1



A 8: Predictions vs Actual plots of store WI_2



A 9: Predictions vs Actual plots of store WI_3