RICE EMS CAPSTONE FINAL REPORT

Authors: Chelsea Zhao & Zoe Katz & Bill Huang & Shenyuan Wu & Yunli Su & Kevin Cai
Rice University
{xz86, zrk1, yh85, sw80, ys125, qc15}@rice.edu

CONTENTS

1 INTRODUCTION

1.1 BACKGROUND

Rice Emergency Medical Services (REMS) serves the Rice community and provides them with accessible medical care. After receiving a phone call, REMS dispatches staff to the site of the request. REMS staff then contact the patient and decide the best treatment plan, which could include calling an ambulance for transport to a nearby hospital. Composed mainly of undergraduate volunteers, REMS responds to about 1,000 emergencies each year and provides 5 academic courses in the Department of Kinesiology.

Since its establishment in 1996, REMS has seen significant increases in student enrollment, call volume (the number of calls), and its number of staff. Due to an expected addition of around 700 students to the student body in the next 3-5 years, Rice plans to build two new residential colleges to accommodate demand.

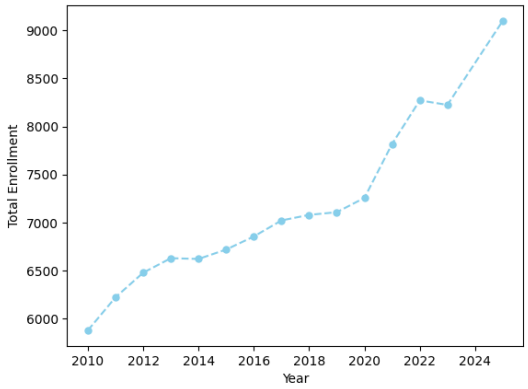

Figure 1: Total Student Enrollment from 2010-2024

REMS would like to predict expected call volume for the next 3-5 years, since this would allow them to be more prepared for the amount of future calls. Call volume can be influenced by a number of factors, such as the number of special events and the number of students living on campus.

Requests for increased budget occur around May each year for a fiscal year that begins on July 1st. Through predicting call volume, personnel need, and supply requirements, REMS can create an accurate and thorough budget forecast. More efficient financial allotment means that REMS can foster a safer and healthier Rice campus.

In this project, we will help REMS prepare for their future as a medical service that provides aid to the Rice community by predicting the factors detailed above. REMS will be better equipped to respond to a variety of emergencies by hiring adequate staff, owning proper equipment, and maintaining rigorous training schedules for their volunteers. With the help of our project, REMS can improve its future health as a service, and, simultaneously, help protect our own.

1.2 OBJECTIVES

To solve the aforementioned problems, the project has been divided into three main sub-objectives. These points, listed below, specify the final goals the project would like to achieve.

- Use past data on call volume to predict the future number of emergency and special event calls.

- Use past data on personnel, including number of available volunteers and paid staff, education and training hours, and housed v.s. off-campus members, to predict future staff growth in all areas, as well as future training needs.
- Use financial information, including the top five line items in primary operations and educational budgets to predict future expenditures.

2 LITERATURE REVIEW

2.1 TIME SERIES INTRODUCTION

Data collected over regular intervals of time is called time-series data and each data point is equally spaced over time. Time series prediction is the method of forecasting upcoming trends or patterns of the given historical dataset with temporal features (Chimmula & Zhang, 2020). Some common time series analysis or predictions include weather predictions, heart rate monitoring, and stock trading.

Given the time series, it is broadly classified into 2 categories i.e. stationary and non-stationary. A series is said to be stationary if it does not depend on the time components like trend, or seasonality effects. Mean and variances of such series are constant with respect to time. Stationary time series is easier to analyze and results skillful forecasting. A time series data is said to be non-stationary if it has trend, seasonality effects in it and changes with respect to time. Statistical properties like mean, variance, standard deviation also change with respect to time (Chimmula & Zhang, 2020).

A common test for stationarity of a time series data is called the Augmented Dickey-Fuller (ADF) test. The ADF test verifies the following null hypothesis: there is a unit root present in a time series. The alternate hypothesis is that there is no unit root, and therefore the time series is stationary. The result of this test is the ADF statistic, which is a negative number. The more negative it is, the stronger the rejection of the null hypothesis. In other words, if the p-value corresponding to the ADF statistic is less than 0.05, we can also reject the null hypothesis and say the series is stationary (Peixeiro, 2022).

If a time series is said to be non-stationary, one simple transformation we can apply in order to achieve stationarity is differencing, which we calculate the change of the output from one timestamp to another (usually the previous timestamp):

$$y'_t = y_t - y_{t-1} \tag{1}$$

This can be useful as it stabilizes the mean of the data (Peixeiro, 2022). A lot of time series models assume stationarity, making it a good practice to transform a non-stationary time series into a stationary one.

2.2 TIME SERIES MODELS

One of the common methods used for time series forecasting is autoregressive (AR) process. The AR model is intuitively more appealing than other models because it describes how an observation directly depends upon one or more previous measurements plus white noise (Kadri et al., 2014):

$$y_t = C + \phi_1 y_{t-1} + \phi_2 y_{t-2} + ... + \phi_p y_{t-p} + \epsilon_t \tag{2}$$

or we can rewrite it as:

$$y_t = C + \sum_{i=1}^{p} \phi_i \cdot y_{t-i} + \epsilon_t, \tag{3}$$

where C is a random constant, y_t is the observation at time t, p is an integer to denote the order, ϕ_j are non-seasonal AR parameters, and ϵ_t is the zero-mean Gaussian noise where $\epsilon_t \sim$ N(0, σ^2).

Another common method used is the moving average (MA) model. MA model describes how an observation depends upon the current white noise term as well as one or more previous errors (Kadri et al., 2014). An MA model is defined as:

$$y_t = \mu + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + ... + \theta_q \epsilon_{t-q} \tag{4}$$

or it can be rewritten as:

$$y_t = \mu + \epsilon_t + \sum_{j=1}^{q} \theta_j \cdot \epsilon_{t-j} \tag{5}$$

where μ is the mean of the series, ϵ_t is the error term at time t, q is an integer to denote the order, and θ_j are non-seasonal MA parameters.

ACF stands for the autocorrelation function. Autocorrelation measures the linear relationship between lagged values of a time series. Thus, the ACF reveals how the correlation between any two values changes as the lag increases. Here, the lag is simply the number of timestamps separating two values. On the other hand, PACF stands for the partial autocorrelation function. The partial autocorrealtion measures the correlation between lagged values in a time series when we remove the influence of correlated lagged values in between. Those are known as confounding variables. The partial autocorrelation function will reveal how the partial autocorrelation varies when the lag increases (Peixeiro, 2022).

If we plot the ACF from some data and see an abrupt change from significant coefficients to insignificant ones until certain lag, then that lag would be the q for our MA model (Peixeiro, 2022). For example, we can conclude that we have a moving average model of order 2, or MA(2) from Figure 2.

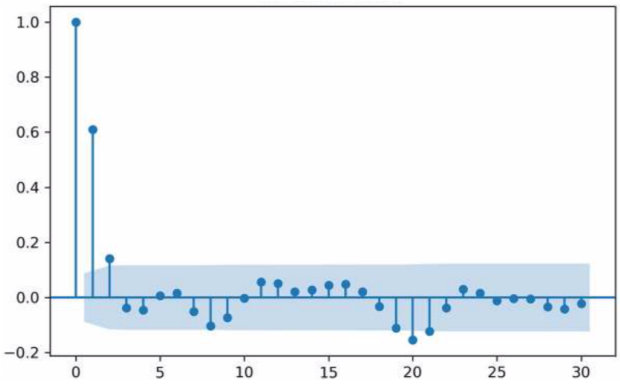

Figure 2: Autocorrelation graph

On the other hand, if we do not see an abrupt change in the ACF plot, we can look at the PACF plot instead. PACF measures the correlation between the lagged values in a time series when we remove the influence of correlated lagged values in between (Peixeiro, 2022). For example, from Figure 3, we can conclude that we have an autoregressive process of order 2, or AR(2)

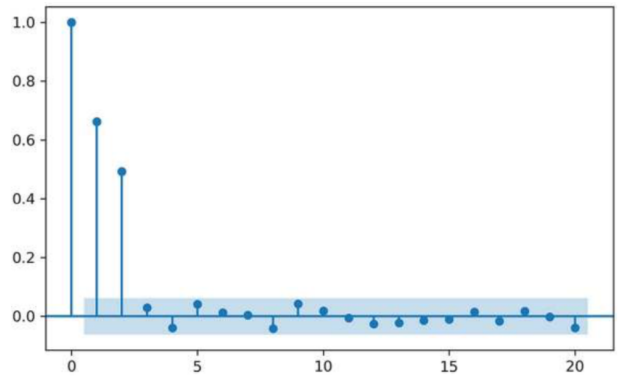

Figure 3: Partial Autocorrelation graph

In other words, we can determine the order for the AR model by looking at the partial autocorrelation function (PACF), and the order for the MA model by looking at the autocorrelation function (ACF).

The ARMA model was introduced (Box and Jenkins) to incorporate both the AR model and the MA model. The ARMA model of order (p,q), or ARMA(p,q), can be written as:

$$y_t = C + \phi_1 y_{t-1} + \phi_2 y_{t-2} + ...\phi_p y_{t-p} + \mu + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + ... + \theta_q \epsilon_{t-q} \tag{6}$$

This can also be rewritten as:

$$y_t = C + \sum_{i=1}^{p} \phi_i \cdot y_{t-i} + \mu + \epsilon_t + \sum_{j=1}^{q} \theta_j \cdot \epsilon_{t-j} \tag{7}$$

Kadri et al. (2014) applied these models in order to predict emergency department overcrowding in France. The patients were divided into six different categories (G1, G2, ... G6) based on their admission mode to the hospital. Three categories were used in the prediction as they account for the majority of the patients, namely G2, G4, and the total number of patients. The ARMA model was applied to all three catogories, and it turns out that ARMA(1,1) was the best model for G4, and ARMA(2,1) was the best model for both G2 and the total number of patients. The models were selected and validated using $RMSE$ and R^2. $RMSE$ is defined as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (\hat{Y}_i - Y_i)^2}{n}} \tag{8}$$

where $\hat{Y}_i$ are the predicted value, Y_i are the observed values, and n is the sample size. R^2 is defined as:

$$R^2 = 1 - \frac{SSR}{SSY} \tag{9}$$

In this equation, SSR is the sum of squared residuals, and SSY is the sum of the squared differences between the observed values and the average observed values:

$$SSR = \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2 \tag{10}$$

$$SSY = \sum_{i=1}^{n} (Y_i - \bar{Y})^2 \tag{11}$$

where Y_i are the observed values, $\hat{Y}_i$ are the predicted values, and $\bar{Y}$ is the average of the observed values. The models selected resulted in low $RMSE$ and high R^2, indicating the models are good representations of the data. Afterwards, the residuals were examined to ensure they follow a Gaussian distribution. At the end, the errors were computed between the observed values and the predicted values, and all three models provide reasonable description and representation of the patient admission to the emergency department.

Previously mentioned models, AR(p), MA(q), and ARMA(p,q), can only be used for stationary time series data. When we encounter non-stationary time series data, transformation like differencing (mentioned above) can be applied. When we do add this extra step in the process, we are also adding a new component to our time series models which leads to the ARIMA model. ARIMA stands for autoregressive integrated moving average. The AR and MA components are the same as the ARMA model; the integrated component here stands for the integration order or differencing order, meaning the number of times it takes to difference the series in order to make it stationary. This integration component is usually denoted by the variable d. ARIMA model can thus be denoted as (p,d,q) where each variable corresponds to each component of the model. The mathematical expression for the ARIMA model is as follows:

$$y'_t = C + \phi_1 y'_{t-1} + ... + \phi_p y'_{t-p} + \theta_1 \epsilon_{t-1} + ... + \theta_q \epsilon_{t-q} + \epsilon_t \tag{12}$$

Here in the equation, y'_t stands for the present value of differenced series at time t, and y'_{t-p} are the past values of differenced series at time $t - p$, and all the other parameters are the same as the ARMA model. We do see that the variable d is not explicitly expressed in the equation here, as the process of making the series stationary will be performed before the model is applied.

In other words, ARIMA model is an ARMA model that can be applied on non-stationary time series data. Because of the fact that ARIMA model can be applied on non-stationary data, it makes it one of the more popular and used time series models. Ariyo et al. (2014) used ARIMA models with different parameters in order to predict several different stock indexes. Nochai & Nochai (2006) used an ARIMA(2,0,1) model to predict the oil palm farm price. Fattah et al. (2018) used an ARIMA(1,0,1) model to predict the demand in a Moroccan food company in order to optimize inventory management. Chen et al. (2008) used an ARIMA(1,0,0) model to predict short-term forecast of property crime in a city in China in order to help local police and governments in decision making and crime suppression.

In addition to ARIMA model, SARIMA model is introduced to account for seasonality. SARIMA stands for seasonal autoregressive integrated moving average, which adds the seasonality component to the ARIMA model. SARIMA model is denoted as SARIMA(p,d,q)(P,D,Q)$_m$, where p, d, and q are all the same variables as the ones in ARIMA model, and the additional parameters P, D, Q are their respective seasonal counterparts. The new parameter m stands for frequency, meaning the number of observations of each cycle. For example, if the data is recorded every month, then we would have 12 observations per cycle, and m would be 12 in this case. Similarly, if the data is recorded quarterly, then we would set m equal to 4. For parameters P, D, and Q, we would include the corresponding past values of the series at a lag that is the multiple of m. For example, if we have a data that is recorded monthly, then we would have m equal to 12, and if we set P as 2, then we would include the past values y_{t-12} and y_{t-24}; similarly if we set Q as 2, then we would include the error terms ϵ_{t-12} and ϵ_{t-24}. If we set D as 2, then we would have a seasonal differenced term with corresponding past values:

$$y'_t = (y_t - y_{t-12}) - (y_{t-12} - y_{t-24}) \tag{13}$$

SARIMA model is particularly useful when the data shows patterns with a certain number of observations. Martinez et al. (2011) developed a SARIMA(2,1,2)(1,1,1)$_{12}$ model to predict the number of dengue cases in Brazil. Deretić et al. (2022) used a SARIMA(0,1,2)(1,1,0)$_{12}$ to predict the number of traffic accidents in Belgrade. Cong et al. (2019) implemented a SARIMA(1,0,0)(0,1,1)$_{12}$ to predict the number of inflenza cases in China. In all cases, these data showed seasonality, and the SARIMA model turned out to be the best model.

All the models that are mentioned above only consider the variables from the time series itself. However, there will be times when other external variables that may affect the target variable, so that they should be considered in the equation. ARIMAX and SARIMAX were introduced to include external variables. The letter "X" in the two models stands for exogenous variables, meaning we can add other input variables in order to make predictions for the target variable. The ARIMAX model can be expressed as:

$$y_t = ARIMA(p,d,q) + \sum_{i=1}^{n} \beta_i X_{i,t} \tag{14}$$

Similarly, the SARIMAX model can be express as:

$$y_t = SARIMA(p,d,q)(P,D,Q)_m + \sum_{i=1}^{n} \beta_i X_{i,t} \tag{15}$$

In the two equations above, $X_{i,t}$ stands for external variables X_i at time t. Aji et al. (2021) used an ARIMA(6,1,4) with the addition of the Google trend of COVID cases as an external variable to forecast the number of COVID-19 cases in Indonesia. Wangdi et al. (2010) implemented a SARIMAX model, a SARIMA(2,1,1)(0,1,1)$_{12}$ model with the addition of the number of cases of malaria in a previous month, mean maximum and minimum temperatures, relative humidity and rainfall lagged at one month as external variables in order to predict the future number of malaria cases in several seven malaria endemic districts.

So far we have mentioned ARMA, ARIMA, SARIMA, ARIMAX, and SARIMAX. These models use one linear regression equation in order to make their forecasts. However, when we have to deal

with time series data with more than one variables, we have to rely models with higher dimensions. The Vector Autoregression, or short for VAR, is a model that is used for multivariate time series. The VAR model is a natural extension of the univariate autoregressive model to dynamic multivariate time series. Since it is possible that two time series have a bidirectional relationship, meaning that one time series is a predictor for the other one. In such a case, it would be useful to have a model that can take this bidirectional relationship into account and output predictions for both time series simultaneously. The VAR model allows us to capture the relationship between multiple time series as they change over time. That, in turn, allows us to produce forecasts for many time series simultaneously, therefore performing multivariate forecasting (Peixeiro, 2022).

The VAR model can be seen as a generalization of the AR(p) model, which is why it can also be denoted as VAR(p). Here, p is the order and it has the same meaning as in the AR(p) model. For a VAR model, we can simply extend an AR(p) model to allow for multiple time series to be modeled, where each variable has an impact on others. Similar to an AR(p) model, VAR requires all time series included to be stationary. For simplicity, we will consider a system with two time series (bivariate), denoted as $y_{1,t}$ and $y_{2,t}$, and an order of one (p=1). Then using matrix notation, the VAR model (VAR(1)) would be expressed as:

$$\begin{bmatrix} y_{1,t} \\ y_{2,t} \end{bmatrix} = \begin{bmatrix} C1 \\ C2 \end{bmatrix} + \begin{bmatrix} \phi_{1,1} & \phi_{1,2} \\ \phi_{2,1} & \phi_{2,2} \end{bmatrix} \cdot \begin{bmatrix} y_{1,t-1} \\ y_{2,t-1} \end{bmatrix} + \begin{bmatrix} \epsilon_{1,t} \\ \epsilon_{2,t} \end{bmatrix} \tag{16}$$

When we multiply the matrix out, we would have the following equations:

$$y_{1,t} = C_1 + \phi_{1,1} y_{1,t-1} + \phi_{1,2} y_{2,t-1} + \epsilon_{1,t} \tag{17}$$

$$y_{2,t} = C_2 + \phi_{2,1} + y_{1,t-1} + \phi_{2,2} y_{2,t-1} + \epsilon_{2,t} \tag{18}$$

As we can see, in the first equation, we included the past values of $y_{2,t}$ in the expression of $y_{1,t}$, and we included the past values of $y_{1,t}$ in the expression of $y_{2,t}$. In this case, the model is able to capture the impact of different variables between each other. A more general model for a VAR(p) model would include up to p lags in the equations as well as the matrix.

The VAR model would be particular useful when we suspect that there exists causal effects between multiple variables within a time series data. Khan et al. (2020) utilized a VAR model to study the relationships and forecast the number of new cases, deaths, and recovery cases from COVID-19 in Pakistan. Gholamzadeh & Bourbour (2020) included different variables to forecast the air pollution in Tehran city with a VAR(6) model. Xiumei et al. (2011) implemented a VAR model to study the relationship between economic growth and carbon emission in certain resource-dependent cities.

Although the statistical methods described above are all popular time series forecasting models, they are all regression-based models. Because of their regression-based nature, it can be difficult for them to predict data with nonlinear relationships between parameters or prediction with longer periods. In this case, machine learning models can be used to work with more complex time series or data with longer periods.

One popular type of model that is used for complex data in machine learning is neural network. A neural network is a method in artificial intelligence that mimics the way human brains work. In human brains, there are neurons that transmit and receive signals to other neurons. By communicating with other neurons, it allows us humans to perform basic but important functions in life. Neural networks aim to capture these characteristics by using interconnected nodes or neurons between layered structures. We can see in Figure 4, a neural network consists of an input layer, some hidden layers, and an output layer. The input layer consists of the data we are given, or the independent variables; the hidden layers are the ones that apply mathematical computation in order to better analyze the data, each hidden layer receives some output from the previous layer, make some calculation or analysis, and passes it on to the next layer; the output layer gives us the prediction for our target variable.

Neural networks can be more optimal than statistical models because they are capable of learning and capturing nonlinear and more complex relationships between the input and output.

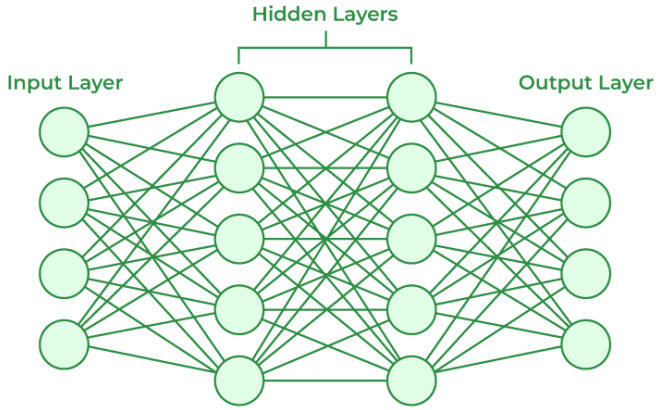

Figure 4: Neural Network

There are different types of neural networks that are built for different tasks. For example, the ones that are used for image processing are called Convolutional Neural Network (CNN); the ones that are used for speech recognition or time series applications are called Recurrent Neural Network (RNN). For the purpose of the first objective of this project, we will use RNN because the data is in the form of time series. Specifically, we will be using a model called Long Short-Term Memory (LSTM) which is a special type of RNN.

Recurrent Neural Networks are a form of neural networks that display temporal behavior through the direct connections between individual layers (Karim et al., 2019). There are several variations of RNN-based models. Most of these RNN-based models differ mainly because of their capabilities in remembering input data... A speical type of RNN models is the Long Short-Term Memory networks, through which the relationships between the longer input and output data are modeled. These RNN-based models, called feedback-based models, are capable of learning from past data, in which several gates into their network architecture are employed in order to remember the past data and thus build the prospective model with respect to the past and current data (Siami-Namini et al., 2019).

In theory, RNNs are able to leverage previous sequential information for arbitrary long sequences. In practice, however, due to RNNs' memory limitations, the length of the sequential information is limited to only a few steps back (Siami-Namini et al., 2019). Long short-term memory (LSTM) addresses this problem by integrating gating functions into their state dynamics (Karim et al., 2019). In general, an LSTM model consists of three gates: forget, input, and output gates. The forget gate makes the decision of preserving/removing the existing information, the input gate specifies the extent to which the new information will be added into the memory, and the output gate controls whether the existing value in the cell contributes to the output (Siami-Namini et al., 2019).

Chimmula & Zhang (2020) applied the LSTM network in order to predict the transmission of COVID-19 in Canada. Gates were used with the help of sigmoid function as the activation function since only positive values should be passed by gates to get clear output. Figure 5 shows an internal diagram of LSTM that was used.

The equations for the 3 gates are:

$$J_t = \texttt{sigmoid}\left(w_J[h_{t-1}, k_t] + b_J\right) \tag{19}$$

$$G_t = \texttt{sigmoid}\left(w_G[h_{t-1}, k_t] + b_G\right) \tag{20}$$

$$P_t = \texttt{sigmoid}\left(w_P[h_{t-1}, k_t] + b_P\right) \tag{21}$$

where J_t, G_t, P_t are the functions of input gate, forget gate, and output gate respectively; w_x are the coefficients of neurons at gate(x), h_{t-1} is the result from previous time stamp, k_t is the input to

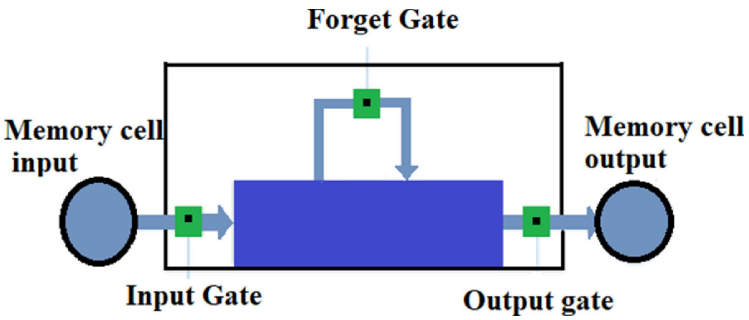

Figure 5: LSTM gates

the current function at time stamp t, and b_x is the bias of neurons at gate(x). Two LSTM models were trained in order to predict both short-term and long-term COVID-19 transmission, the first one on the data from Canada, and the second one on the data from Italy. For the first model that was trained on data from Canada, the RMSE was 34.83 and short-term prediction and 45.7 for long-term prediction. For the second model that was trained on data from Italy, the RMSE was about 51.46. Both models were relatively accurate, with an accuracy score of above 90% for both of them. The models were able to serve as useful tools for Canadian government to monitor the situation regarding COVID-19 transmission.

Siami-Namini et al. (2019) introduced the method of deep-bidirectional LSTMs (BiLSTM), which is an extension of the LSTM model. In a deep-bidirectional LSTM model, two LSTMs are applied to the input data. In the first round, an LSTM is applied on the input sequence (i.e., forward layer). In the second round, the reverse form of the input sequence is fed into the LSTM model (i.e., backward layer). Applying the LSTM twice leads to improve learning long-term dependencies and thus consequently will improve the accuracy of the model (Siami-Namini et al., 2019). It has been reported that using BiLSTM models outperforms regular LSTMs. Siami-Namini et al. (2019) were able to apply both LSTM and BiLSTM on a financial time series data and show that BiLSTM model outperforms LSTM model by 37.78% reduction in error rates.

The model mentioned above, including the time series models and the neural networks, work well with relatively large dataset. When we have smaller datasets, simpler regression models could be used in order to avoid the problem of overfitting. Some examples include linear regression model, polynomial regression model, k nearest neighbors (kNN), and so on. For objective two and three, we have very limited data and we could potentially achieve better results with these simpler models. Ban et al. (2013) used a kNN method to do financial forecasting on 121 stocks from S&P500. Al-Qahtani & Crone (2013) was able to develop a multivariate kNN model in order to forecast the electricity in the UK market.

All the models mentioned in this section are potential candidates for our model selection. Selective models that are deemed appropriate will be fit for each specific objective, and the most optimal model for each objective will be selected. The more detailed processes of model selection will be discuss in the following sections.

3 Data Description

3.1 Tabular Call Summary and Description

The call data is presented through 2 files that have the same features and records the call history from January 2006 to April 2018, and from May 2018 to December 2023, respectively. It is obtained directly from REMS records. The dimension after aggregation of the data (under the same features) is 11,943 x 8, with 11,943 calls recorded and 8 features for each call.

Table 1: Tabular call data description

	Dataset	Rows	Columns	Size
1	Calls [Jan 2006 - Apr 2018]	8284	8	882 KB
2	Calls [May 2018 - Dec 2023]	3659	8	213 KB

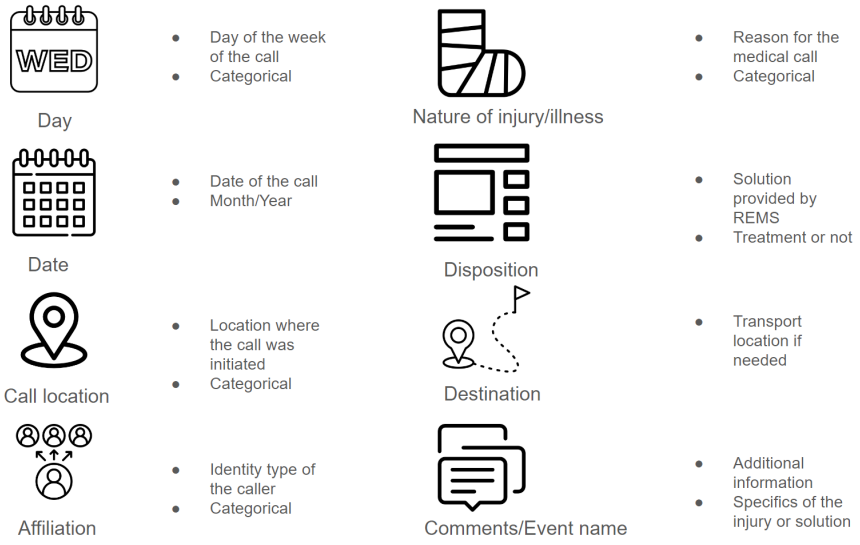

Figure 6: Tabular feature description for the call descriptions. The features specify the time and location of the call, as well as the caller information and the subsequent management.

3.2 Tabular EMS Excel White Paper

The white paper data consists of 14 tabs that are extracted and modified by the team to qualify for data analysis into 23 files. Figure 7 visualizes the data structure. It is made up of 3 main types of information: budget and costs, personnel and enrollment, and inventory and supplies. Call volume, enrollment, and date are the most important features in the analysis to predict call numbers and forecast budget requirements on a monthly basis. The data is limited in the sense that the time frame over which different data is available do not match (for example, the number of calls is available from 2006 to 2023 on a monthly basis, yet the staff number is only available from 2016 onwards).

Table 2: Tabular EMS white paper

	Dataset	Rows	Columns	Size
1	Budget Forecast	8284	8	882 KB
2	Call Volume	15	9	2 KB
3	Call Outcomes Type	14	8	1 KB
4	Call Patients Type	14	6	1 KB
5	Cost Each Medical Bag	1	4	1 KB
6	Event Hours	39	5	2 KB
7	Event Number	12	4	1 KB
8	Golf Cart Inventory	5	4	28 KB
9	Golf Cart Maintenance	22	5	30 KB
10	Golf Cart Usage	23	6	30 KB
11	Medical Gear Plan 2023	7	12	1 KB
12	Medical Gear Usage 2023	11	11	1 KB
13	Monthly Employee Enrollment	108	2	2 KB
14	Monthly Calls	216	2	3 KB
15	Monthly Enrollment	162	4	4 KB
16	Patient Type	13	6	1 KB
17	Off-campus Room	84	5	30 KB
18	Radios	52	4	31 KB
19	Special Event Academic Year	13	3	1 KB
20	Special Events 2006-2023	216	2	3 KB
21	Staffing	9	7	1 KB
22	Staff Number Change	14	12	1 KB
23	Volunteer Hours	25	6	29 KB

3.3 Annual / Academic year report PDFs

27 annual and academic year report PDFs were prepared to highlight the achievements from the year as well as summarize the state of the REMS. Largely serving as summaries of the tabular data, there are some new information that the team manually extracted, such as the staff composition data and education hours data. Figure 9 displays snapshots of the PDF data.

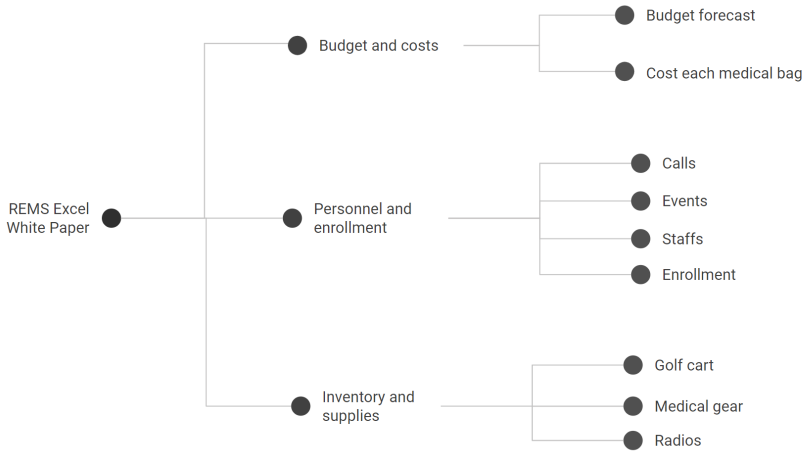

Figure 7: Tabular feature description for the white paper. The datasets are grouped to reflect different types of information they carry.

- Call records: 2006 – 2023
- Student enrollment: 2006 – 2023
- Employee enrollment: 2016 – 2023
- Event hour data: 2021 – 2023

Figure 8: Time range mismatching in the tabular data

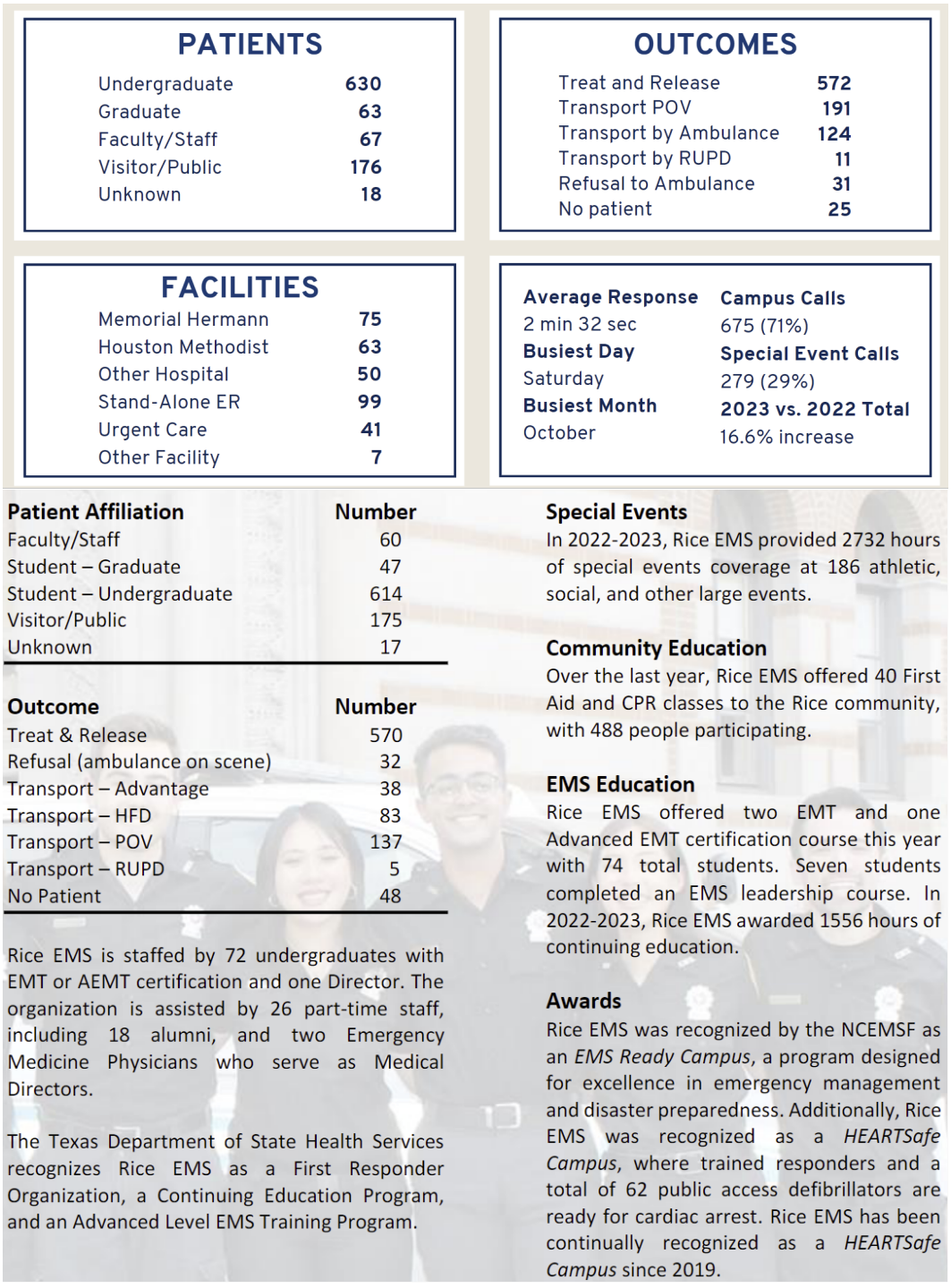

Figure 9: Yearly report snapshots

4 Data Science Pipeline

The data science pipeline includes three main components: data wrangling, data exploration, and modeling. This section involves the details of data wrangling and data exploration, as well as the explanation of the models that will be applied for Objective 1: Predicting the future number of emergency calls, Objective 2: Predicting the future number of staff and training needs, and Objective 3: Predicting five kinds of expenditures in the future.

4.1 Objective 1

4.1.1 Data Wrangling

There are three main tasks to deal with during the stage of data wrangling for Objective 1: how to convert yearly data into monthly data, how to process missing values, and how to merge all the data into one dataset.

Instead of predicting yearly call volume with the help of yearly values of features, call volume and other features recorded per month are used due to several reasons. First, REMS began recording call volume in 2006. Since the data ends in 2023, the data REMS has only spans 18 years (in other words, this is 18 rows of yearly data). This is a very small data size; therefore, it is difficult to predict yearly call volume precisely no matter what models are used. Since call volume and the number of special events change every month, it is meaningful to enlarge the dataset by using monthly data. With a large dataset where the combination of data in each row almost never repeats, applied models could get a much more precise prediction. In addition, predicting monthly call volume is more meaningful than predicting yearly call volume from the perspective of REMS. Calls do typically increase and decrease based on which month it is; this means it would be helpful for REMS to know which time periods they should expect more calls. Compared to knowing the approximate number of call volume in the next year, knowing the approximate number of call volume in several following months in advance could help REMS prepare for those months better.

In order to acquire a dataset containing monthly data, we extracted monthly call counts from the two excel files that record information of each call from 2006 onward. For data such as the number of undergraduate enrollment and the number of employees is recorded in a yearly or per-semester basis. Therefore, their values are quite consistent throughout a year/semester and can be filled as a static number based on the semester the month belongs to. Similarly, the number of employees per month is calculated based on the year it belongs to. For example, the number of student enrollment in September 2023 will be extracted by the corresponding value of student enrollment in Fall 2023.

Since REMS does not have data recording the number of people staying on campus at Rice in any summer, we are not able to impute the values in those months. After discussing with the sponsor, we decided not to predict the call volume in future summers. This is mainly due to both a significantly reduced summer call volume (typically around 20 calls per month) as well as because REMS does not have strong desire to predict the call volume in the summer months. Their focus is much more on the academic school year, which runs from August to April. Therefore, we chose not to include data in May, June, and July. Figure 10 illustrates the call volume in each month, where there is a noticeable decline in calls from May to July.

We do not have data on the number of employees from 2006 to 2015. Although we lack these 10 years of employee data, we cannot ignore this feature since its importance cannot be proved without additional testing. Therefore, we currently have two datasets: one that spans 2006-2023 without the employee data and one that spans only 2016-2023 with the employee data. After determining the importance of employee data, we can then decide which dataset to use.

Finally, to merge all the data, the datasets are combined based on the months each row belongs to. However, since the date format in each dataset is different, we first converted all date columns into the same format and then joined the features by matching the values in the new, standardized, date columns.

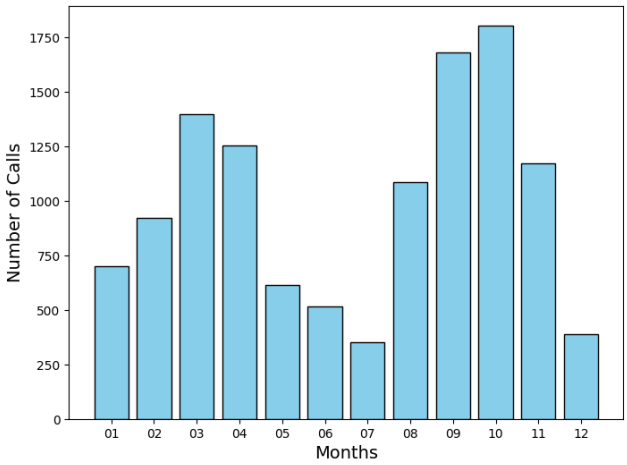

Figure 10: Call Volume by All Months

After all problems related to data wrangling were resolved, we ended up with two candidate datasets as mentioned above for Objective 1. The dataset without employee data has 139 rows and 6 columns. The dataset containing employee data has 72 rows and 7 columns. The variables are defined as follows:

1. **Year-Month**: Each value in the column "Year-Month" represents the month the corresponding row records. In time-series models, this is not considered as a "predictor variable" or a "response variable."
2. **Call Count**: Each value in the column "Call Count" represents the number of calls in the corresponding month, also known as the call volume. This column is the response variable for Objective 1.
3. **Number of SE**: Each value in the column "Number of SE" represents the number of special events in the corresponding month. Special events are large-scale events at Rice, such as public parties, that typically have REMS on site. This column is a predictor variable for Objective 1.
4. **UG Enrollment**: Each value in the column "UG Enrollment" represents the number of non-visiting, full-time undergraduate students studying at Rice in that month, which is a predictor variable for Objective 1.
5. **GR Enrollment**: Each value in the column "GR Enrollment" represents the number of non-visiting, full-time graduate students studying at Rice in that month, which is a predictor variable for Objective 1.
6. **Total Enrollment**: Each value in the column "Total Enrollment" represents the number of all people studying at Rice in that month, including visiting and part-time students, which is a predictor variable for Objective 1.
7. **Number of Employees**: Each value in the column "Number of Employees" represents the number of employees working at Rice in that month, which is a predictor variable for Objective 1.

Table 3 displays the first five rows in the dataset containing employee data to better illustrate the potential final dataset.

Table 4 displays the first five rows in the dataset not containing employee data to better visualize another potential final dataset.

4.1.2 Data Exploration

Year-Month	Call Count	Number of SE	UG Enrollment	GR Enrollment	Total Enrollment	Number of Employees
2016-01	43	10	3724	2596	6437	3519
2016-02	37	16	3724	2596	6437	3519
2016-03	96	24	3724	2596	6437	3519
2016-04	74	23	3724	2596	6437	3519
2016-08	61	7	3839	2861	6855	3519

Table 3: The final dataset containing employee data

Year-Month	Call Count	Number of SE	UG Enrollment	GR Enrollment	Total Enrollment
2006-01	30	7	2888	1892	4924
2006-02	50	21	2888	1892	4924
2006-03	33	16	2888	1892	4924
2006-04	90	12	2888	1892	4924
2006-08	40	4	2959	2013	5119

Table 4: The final dataset not containing employee data

To better understand Objective 1 (Predict the future volume of emergency and special event calls), we analyzed historical call volume data, categorizing and identifying trends over recent years.

Figure 11 shows call volume by various identities that people in the Rice community hold: graduate student, staff, undergraduate student, visitor, and unknown. This figure shows, overall, call volume fluctuates significantly over the past 13 years. For instance, there is a noticeable drop in calls from undergraduate students in 2020, likely due to the COVID-19 pandemic reducing campus presence. This indicates that there may be a relationship between call volume and special events (large-scale public events such as parties and graduation). After the creation of this figure, we were more informed in our understanding of the data. Therefore, we then concentrated on selecting relevant features for predicting call volume.

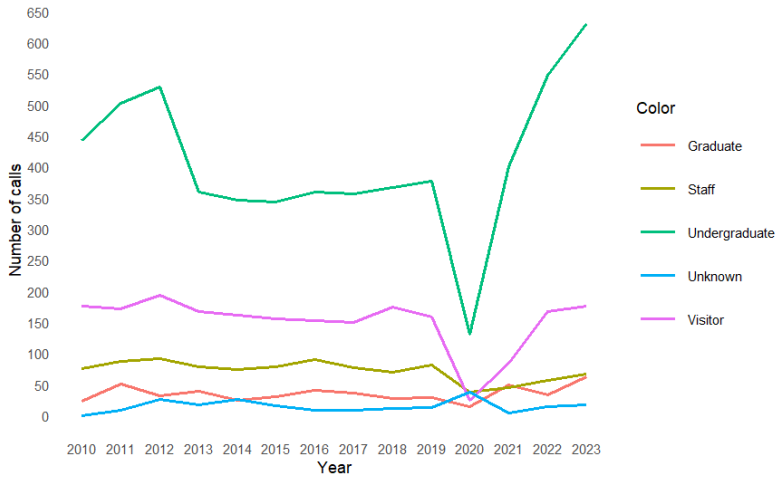

Figure 11: Call Volume Trend by Identity

After analyzing the dataset with five features, it is important to note that the "Number of Employees" data spans only from 2016 to 2023, lacking a decade's worth of data compared to other variables within the entire 2006-2023 dataset. This discrepancy raises concerns about its inclusion. To assess the necessity of this particular feature, we applied Principal Component Analysis (PCA) and evaluated feature importance, excluding target and time variables. Standardization of these variables ensures balanced influence in PCA, and we measured the individual and cumulative percentage of variance explained (PVE) by each principal component. This process revealed in Figure 12 and

Figure 13 that the first two principal components account for a significant 92.58% of data variance, guiding the prediction model's primary factor.

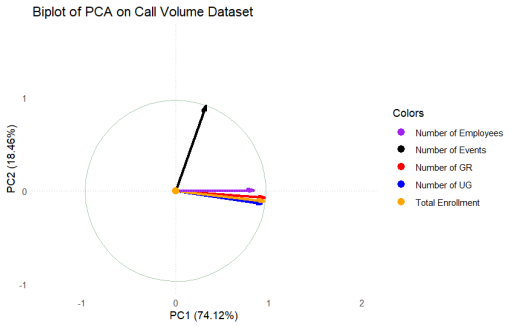

Figure 12: PCA for Objective 1

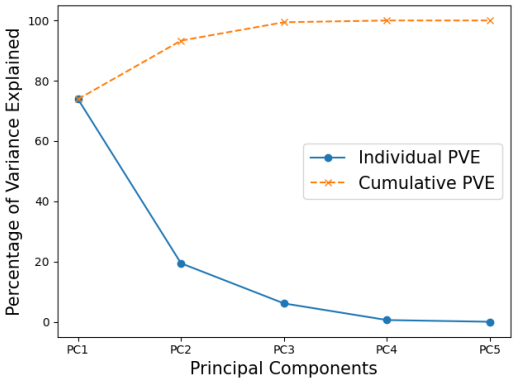

Figure 13: Percentage of Variance Explained by PCA Components

Subsequently, a scatter plot was generated, which is shown in Figure 14, mapping data points to these principal components. The color coding of the points corresponds to the call volume. This visualization highlighted a concentration of high call volume points in the upper right quadrant, supporting PCA's effectiveness in capturing meaningful data patterns.

Based on the value of variance, that is, the slope of each feature in each principal component hyperplane, we created a heat map as shown in Figure 15. This figure shows the values of variance for features in the first two components.

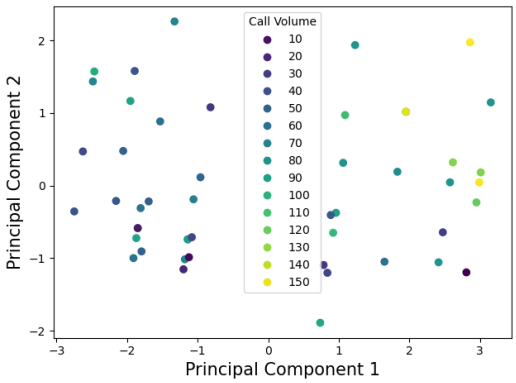

Figure 14: PCA of 2016-2013 Dataset

Figure 15: Heatmap of Feature Loadings on Principal Components

Principal component one is heavily influenced by the following variables: "Undergraduate (full-time, non-visiting)," "Graduate (full-time, non-visiting)," "Total (including part-time, visiting)," and "Number of Employees." Since the loadings for these variables are all positive and range from approximately 0.46 to 0.51, suggesting that they are all moderately contributing to PC1 in a similar direction. Given the strong correlation of "Number of Employees" with other variables shown in Figure 16, the decision was made to proceed with a dataset excluding this variable to maintain dataset integrity and completeness.

The variable "Number of Special Events" demonstrated a strong positive correlation with PC2, marked by a high loading of 0.97. Consequently, excluding employee data, the refined dataset comprises 139 rows and 6 columns, ensuring a more complete and relevant analysis.

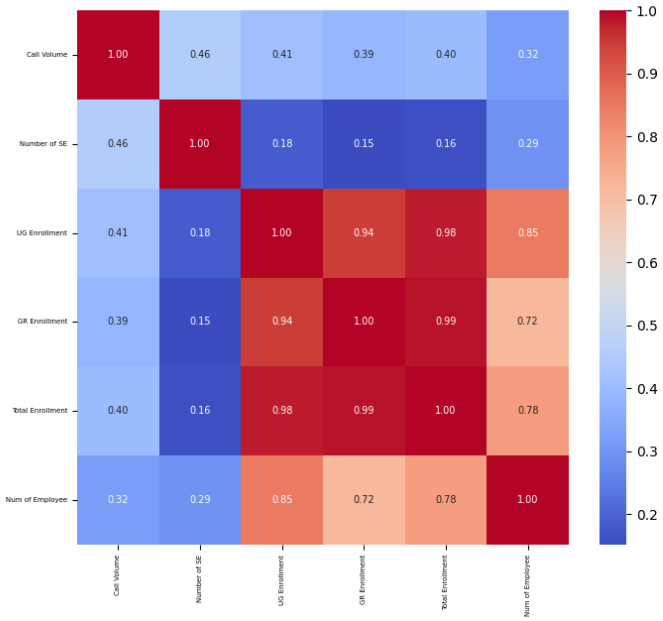

Figure 16: Correlation among features in Objective 1

4.1.3 Modeling

Considering the nature of our dataset and the goals of our project, we can formulate all of our objectives as time series forecasting problems. Different time series models can be applied to each objective depending on the nature and the pattern of our data. The models that we used for this objective include the traditional time series models that were mentioned above.

The first model that will be implemented is the LSTM model. LSTM is type of Recurrent Neural Network (RNN) and it is used to process sequential data. We would like to see if using a neural network could potentially provide an optimal performance because of its architecture.

Some other models that will be used are the statistical time series models that were mentioned above, including ARIMA, SARIMA, ARIMAX, and SARIMAX. We decided to use the ARIMA model as a baseline model. After plotting the call volume, there seems to be a pattern between certain months, so we decided to use SARIMA to examine if there exist seasonlity within the data. We also decided to use ARIMAX and SARIMAX because we think other variables (such as the number of special events in each month) in the dataset could be good predictors of call volume, and we would like to add these factors as external variables.

4.2 Objective 2

4.2.1 Data Wrangling

The key issue in the Objective 2 data is that, although we do have data that covers a variety of features, these features span across different time periods, have different frequencies (yearly and monthly), and have different formats in the original Excel files.

We first extracted all Excel files into several data frames. They are listed and described below:

1. **Event Hours**: After cleaning, the Date column is the way to match this dataframe with the others by month. The other columns cover information including Paid hours, Volun-

teer hours, and Total hours. In time-series models at the end, this is not considered as a "predictor variable" or a "response variable."

2. **Monthly Employee Enrollment**: Each number represents the number of employees during that specific month. The value is static in roughly half-year periods because the staff count remains constant during a semester, as Rice is semester-based. This column is a predictor variable for Objective 2.
3. **Staffing**: The staffs are divided into several types, including IC (in charge), DC (duty crew), UG (undergraduate volunteers), part timers, and graduate volunteers. The yearly data is transformed into monthly in our data wrangling step to match the frequency of the rest of the data. This column is a response variable for Objective 2.
4. **Volunteer Hours**: It gives monthly volunteer hours by each type of staff (IC/DC only), at the end, this is not considered as a "predictor variable" or a "response variable."
5. **Education vs Training Hours**: These are the yearly education hours and training hours for staff hosted by REMS, transformed into monthly to match the common frequency. It is a predictor variable for Objective 2.
6. **Housed vs OC Staff**: The monthly data for number of shifts and hours of down room usage for off-campus on-duty staff provides information for the room usage, which is a response variable for Objective 2.

To organize this data, we made the first column for all data frames to be the date. Then, we removed the NA values and combined the data together by whether it was yearly or monthly collected, so that our resulting dataframes are condensed into two files for the predictions of both total staff count and monthly off-campus room usage.

Finally, we will describe the two final datasets used in the analysis of Objetive 2: Staff Data Model and Staff Room Hours. The first of which, Staff Data Model, was used to predict the number of staff that REMS could expect to have in the future. This is important for budgetary reasons, since some of the REMS staff is paid, as well as for training reasons, since REMS trains their volunteer staff. The Staff Data Model dataset contains the following features:

1. **Year**: This represents which school year the data is taken from.
2. **Volunteer**: This represents the number of volunteer staff on REMS for that school year. Volunteers are primarily undergraduate students who have gone through extensive training.
3. **Paid**: This column represents the number of paid staff on REMS for that school year.
4. **Total**: This column is the total number of staff on REMS for that school year. It can be calculated by adding the Volunteer and Paid columns.

The second dataset, Staff Room Hours, is used to predict the number of monthly hours the off-campus room was used. REMS could use this information to justify needing an increase in budget to purchase a second off-campus room. the Staff Room Hours dataset contains the following features:

1. **Date**: This column represents the month and year that corresponds to the data in the rest of the row.
2. **Hours**: This columns shows the number of hours that the off-campus room was used by REMS staff for that month.
3. **Paid Staff**: This column represents the number of paid staff on REMS for that month. This variable remains constant for each school year.
4. **Volunteer Staff**: This column represents the number of volunteer staff on REMS for that month. This variable remains constant for each school year.
5. **Total Staff**: This column is the total number of staff on REMS for that month. It can be calculated by adding the Volunteer and Paid columns, and it also remains constant for each school year.

4.2.2 Data Exploration

We explore the correlation between the variables in the cleaned dataset first to see which predictors are likely more relevant in our times series model for staff count prediction.

The positive correlations between total staff and hours, and between volunteer staff and hours both make sense. We then observe the staff count given by the dataset over the time frame. Figure 18 shows a generally increasing trend in total number of staff from 2015 to 2024.

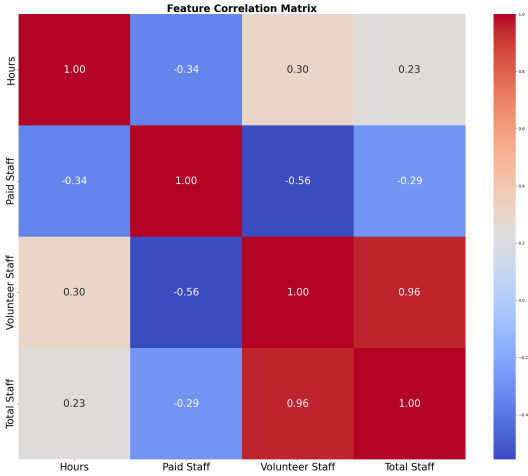

Figure 17: Feature correlation matrix

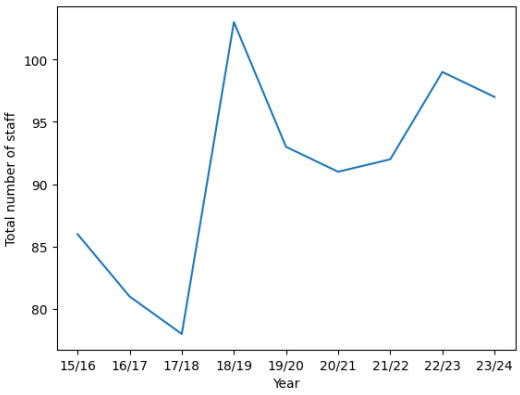

Figure 18: Academic Yearly Staff Count over 2015-2024

To visualize the trend of the other times series, as it will be informational for the VAR model, we graphed the time series of the other variables in our cleaned dataset, as shown by Figure 19.

4.2.3 Modeling

ARIMA models are useful given the strong temporal association between our data for Objective 2.

To predict future staff count, we first check the autocorrelation plot to select the appropriate parameter values for the ARIMA model. If it does not work, we will explore other ways of selecting parameters. Finally, we will fit the model to the historical data and use it to forecast future staff counts, incorporating potential walk-forward prediction due to the low data volume available as the data is yearly, adjusting the model as necessary based on error measures to ensure accuracy and reliability in the predictions.

VAR stands for Vector AutoRegression. Due to the existence of multiple variables in our time series data, each variable may depend not only on its past values but also has some dependency on other variables. Hence we use VAR, a multivariate forecasting algorithm that is used when two or more time series influence each other. The formula for a k-dimensional VAR model is:

$$\begin{aligned} y_{1,t} &= c_1 + \phi_{11,1} y_{1,t-1} + \phi_{12,1} y_{2,t-1} + \epsilon_{1,t} \\ y_{2,t} &= c_2 + \phi_{21,1} y_{1,t-1} + \phi_{22,1} y_{2,t-1} + \epsilon_{2,t} \\ &\dots \\ y_{k,t} &= c_k + \phi_{k1,1} y_{1,t-1} + \phi_{k2,1} y_{2,t-1} + \epsilon_{k,t} \end{aligned} \tag{22}$$

It is modelled as a system of equations with one equation per time series variable. Here k represents the count of time series variables.

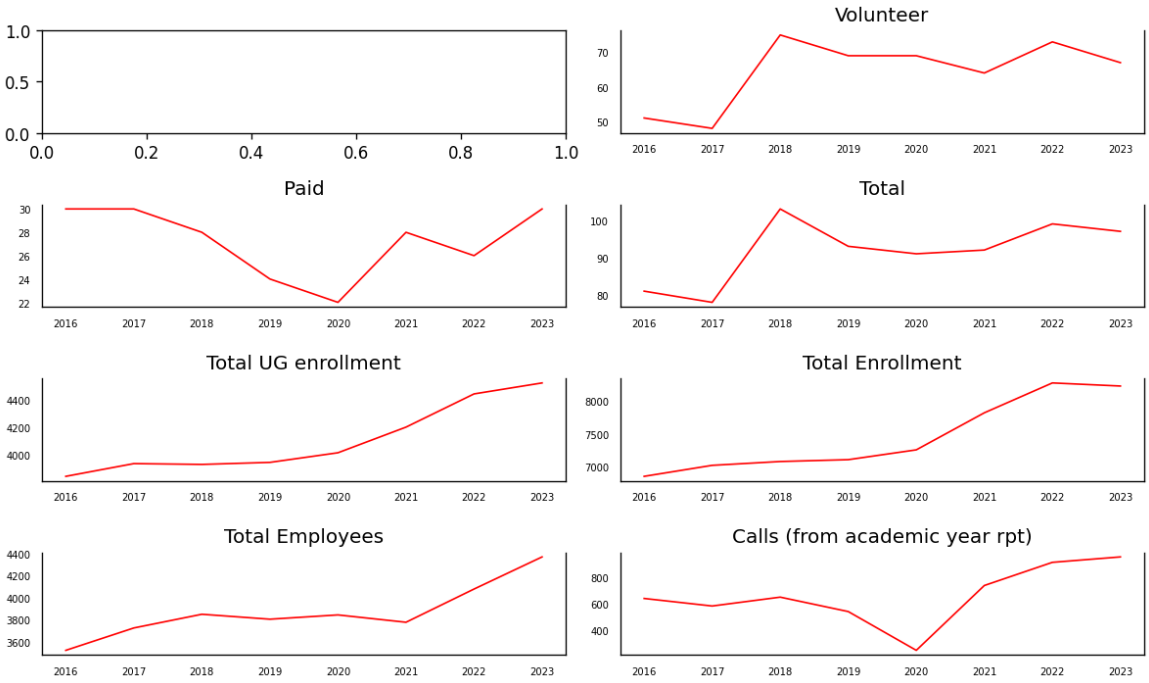

Figure 19: Time series trend for relevant variables

SARIMA models, as described for Objective 1, are useful for data that has seasonal variation. The monthly usage of the OC room varies by season, for instance. In the spring and fall, usage increases, and in the summer and winter, usage declines. Therefore, SARIMA was determined to be a good fit to predict the future monthly usage of the OC room.

For the rigor of data analysis, we considered the appropriateness of applying time series models on the available dataset. With limited data volume (only 8 years of annual data available), there may not be enough data to feed a time series model. Thus, we apply polynomial regression at the same time to compare the results with time series models applied:

$$y =_{+} a_1 x + a_2 x^2 + ... + a_n x^n \tag{23}$$

Polynomial regression has a comparatively lower data volume requirement than time series models.

4.3 Objective 3

4.3.1 Data Wrangling

Objective 3 is to use past data to predict five different aspects of expenditures. The only issue we need to handle is the two empty values, one is the cost of insurance in 2017 and one is the cost of IC housing in 2020. To solve this, we applied the technique K-Nearest-Neighbors to impute these two values since this small dataset does not allow us to lose more data. We set the number of neighbors (K) to be 2, guaranteeing that the imputation does not simply rely on a single data point which might be an ”outlier”. Overall, we extracted features from the data source, containing the following features:

1. **Academic Year**: The academic year.

2. **Uniforms**: The cost of uniform in the corresponding year.
3. **IC Housing**: The cost of IC housing in the corresponding year.
4. **Medical Supplies**: The cost of medical supplies in the corresponding year.
5. **Vehicle Maintenance**: The cost of vehicle maintenance in the corresponding year.
6. **Insurance**: The cost of insurance in the corresponding year.
7. **Total Enrollment**: The number of student enrollment at Rice in the corresponding year.
8. **Total REMS Staff**: The number of REMS Staff in the corresponding year.
9. **Call Volume**: The number of emergency calls in the corresponding year.

Table 5 displays the first two rows and the last row of the dataset.

Academic Year	Medical Supplies	IC Housing	Vehicle Maintenance	Insurance	Uniforms	Total Enrollment	Call Volume	Total REMS Staff
2016	7780.0	10328.0	7918.00	5153.0	7429.0	6855	640	81
2017	9844.0	11880.0	8563.00	6576.5	5557.0	7022	583	78
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
2022	7431.0	15523.0	3786.73	8494.0	5861.0	8272	913	99

Table 5: Your table caption here.

4.3.2 Data Exploration

Figure 20 shows the trend of the expenditures for the top five line items, namely Vehicle Maintenance, IC (In-Charge) Housing, Insurance, Medical Supplies, and Uniforms.

By visualizing the time series trends for these five different types of expenditures over the period from 2016 to 2022, we can choose our time series model or regression model based on the characteristics of the historical data.

4.3.3 Data Modeling

ARIMA is potentially a good model for Objective 3. As mentioned in the literature review, ARIMA has interpretable parameters (e.g., autoregressive, differencing, and moving average parameters) that can provide insights for the prediction. ARIMA is also the best choice for future forecast since we do not need to use the estimate of the future value of other features, which we do not have, when applying ARIMA.

KNN regression is potentially a good model for Objective 3 as well. KNN regression, unlike some regression methods, can handle complex, non-linear relationships in the time series data. Since the predictions are based on the values of the nearest neighbors, this method has intuitive interpretation, which can help us analyze the relationship between each expenditure and other features. In addition, since our dataset currently is small and will not be very large in the near future, this method will be less possible to overfit a lot compared to other regression methods.

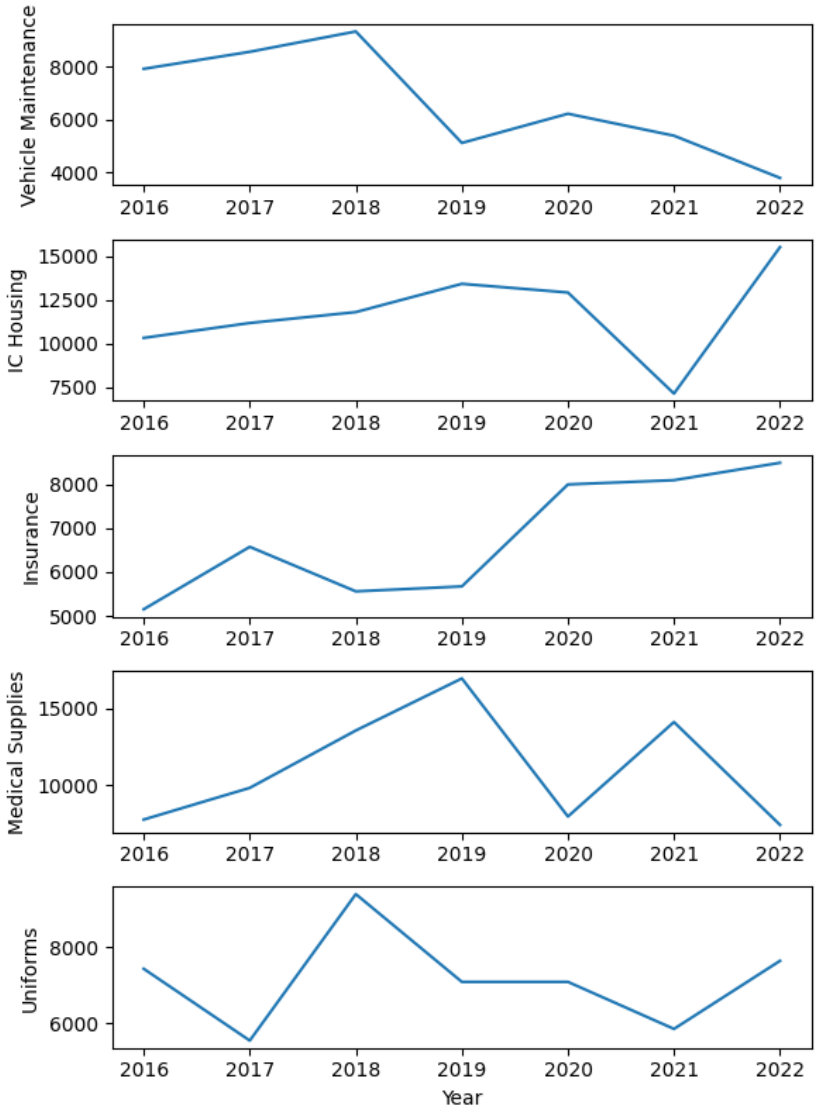

Figure 20: Time series trend for the five expenditures

5 EXPERIMENTS

This section includes the different approaches on how we ran the experiments including how we split our dataset, how we determined the hyper-parameters for our models, and the way we validated our models for three objectives.

5.1 SETUP

The following sections will discuss the methods we planned to use in order to find the most optimal model, including splitting data into training data and testing data, the metrics we used to evaluate the models, and some methods for tuning hyper-parameters.

5.1.1 DATA SPLIT

Given the temporal nature of our datasets, we have partitioned the data into a training set and a testing set in the same way for all three objectives: the training set comprises the initial 80% of the chronological data sequence, which is utilized to train the predictive models. The remaining 20%, representing the last multiple months/years of the time series, is designated as the testing set. The testing set is employed to evaluate the models' predictive performance and to validate their accuracy in forecasting future values.

After the best models are obtained through training and testing, the entire dataset (training set plus testing set) will be trained through the selected models, and forecasts for the next three years will be made for each objective.

5.1.2 MODEL VALIDATION

In order to have the best performance, we need to select the most optimal parameters for the models. We have previously mentioned the method of ACF and PACF plots in order to find the parameters for an ARIMA model. In addition to ACF and PACF, we will also use AIC as an additional method to select the parameters for our models. AIC stands for Akaike Information Criterion, and it is an estimator of prediction error which reflects the quality of the fit of a model on a given dataset. The expression for AIC is:

$$AIC = 2k - 2ln(\hat{L}) \tag{24}$$

where k is the number of estimated parameters in the model and $\hat{L}$ is the maximum value of the likelihood function of the model. The model with a lower AIC score means that it is more likely for this model to minimize information loss, which makes the model a better fit. In other words, we would like our model to have a lower AIC score.

After selecting the most optimal parameters for the models, we will then evaluate our performance by mainly looking at two aspects: performance metrics and graphs showing important information of residuals.

We will look at the training error and the testing error of each model after modeling. The metrics we chose is RMSE. Suppose the number of samples is n and the value of response variable in sample i is denoted by y_i. Hence the predicted value of y_i can be denoted by $\hat{y}_i$. RMSE can be defined as

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \tag{25}$$

We tend to use this metric since it is relatively sensitive to outliers. This satisfies our need since providing REMS outliers might lead to a large waste or deficit of budget. In the mean time, RMSE is expressed in the same units as the response variable being predicted. This interpretable metric helps us understand the average value of prediction error we make.

In addition to performance metrics mentioned above, for Objective 1, we will also check the plot of standardized residuals, the plot of ACF of residuals, and the normal Q-Q plot of standard residuals

to see whether there shows any pattern in residuals. If so, it means we still can extract more useful characteristics of the time series, which might lead to better performance. If the plots show that the standard residuals are distributed normally with a mean of zero and the ACF value at all lag point are small, it means the model has captured all the information in the data, except for the noise inherent in any process.

5.1.3 Model Training

For Objective one, the first model that we implemented was the LSTM model. The hyperparameters for an LSTM model include the number of layers, the number of neurons in each layer, batch size, and epochs. A grid search and trial and error method were applied in order to find the best parameters for the LSTM model. This was done by training each model with different parameters, and comparing each model's prediction with the testing set. The model with the lowest testing error was selected at the end. Our final LSTM model has two hidden layers, where each layer contains 128 neurons. Batch size was set as 1, and epochs was set to be 11.

For the ARIMA model, the most optimal parameters p, d, and q need to be found. The original time series data was not stationary after applying an ADF test, but the data was found to be stationary after differencing once, so the value for d is set to be 1. Figure 21 shows the ACF plot, and figure 22 shows the PACF plot. From both plots, multiple significant lags can be seen throughout the graphs.

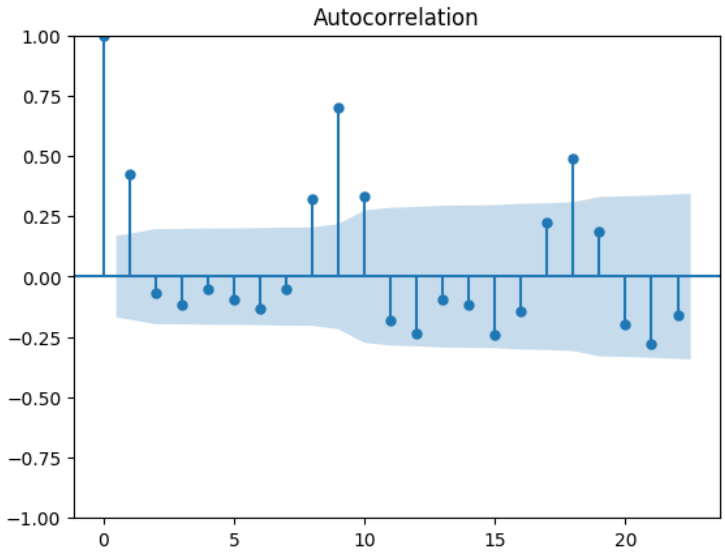

Figure 21: ACF plot

We decided to use grid search in order to find the best parameters. In addition, AIC was calculated for the ARIMA model with every possible combination of the three parameters, and the ones with the lowest AIC was selected. Table 6 shows some of the parameters that resulted in the lowest AIC. From the table, we can see that when p, d, q are set equal to 10, 1, and 0 respectively, the model has the lowest AIC, so we will use these three values for our final ARIMA model.

p	d	q	AIC
10	1	0	923.82
8	1	2	924.46
10	1	1	925.65
11	1	0	925.66
8	1	3	926.19

Table 6: AIC for each ARIMA parameter

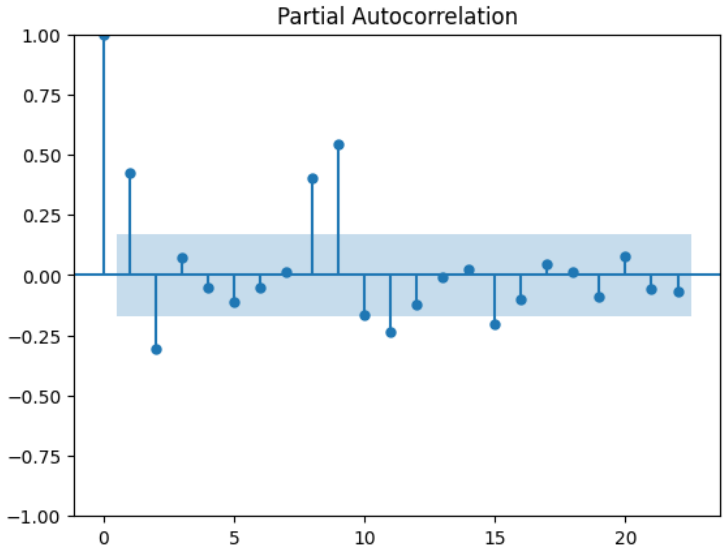

Figure 22: PACF plot

After selecting the parameters and training the model with the selected parameters, we also decided to examine the plot of the residuals from our model. Figure 23 shows the distributions and the correlation of the residuals. As we can see, the top-left plot shows the standardized residuals across the dataset. There does not appear to be any trend, and the mean appears to be stable over the time throughout the plot, which indicates that the data was stationary. The top-right plot shows a histogram of the residuals. We can see that the distribution is a close resemblance of a normal distribution, which indicates that the residuals are roughly normally distributed. The plot on the bottom-left shows the Q-Q plot, which is a plot of the quantiles of two distributions against each other. If the distribution of the residuals is normal, then the Q-Q plot would show a straight line of 45 degrees, which is also the line corresponding to y = x. From our Q-Q plot, we see that the points do form a straight line, which is also an indication of the normality of the residuals. Last but not least, in the bottom right the plot shows the autocorrelation function of the residuals. Since we only see a significant lag that peaks at lag 0, it means that the residuals are not correlated. All the plots are strong indications of the normality and randomness of the residuals, which means that our residuals are strong resemblance of white noise, making this model a good fit for our data.

Similar process was applied in order to find the most optimal parameters for the SARIMA model. Since the data was found to be stationary after differencing once, the value for d and D are set to be 1. In addition, since we took out the 3 months for the summer for each year from the dataset, the frequency would be 9 in our case (9 months per year). The results can be seen in table 7. The final SARIMA model to be selected has a value of 0, 1, 1 for p, d, q respectively, and a value of 1, 1, 1 for P, D, Q respectively, and a value of 9 for m.

p	d	q	P	D	Q	AIC
0	1	1	1	1	1	833.61
0	1	2	1	1	1	835.38
1	1	1	1	1	1	835.39
0	1	1	0	1	1	836.77
2	1	1	1	1	1	837.30

Table 7: AIC for each SARIMA parameter

Residual plots are also plotted for the SARIMA model. Figure 24 shows the four plots. As we can see, these plots show similar patterns as the ones from the ARIMA model. From the top-left plot, we see that the residuals are relatively constant throughout the dataset; from the top-right plot, the distribution of the residuals shown is very close to a normal distribution; from the bottom-left

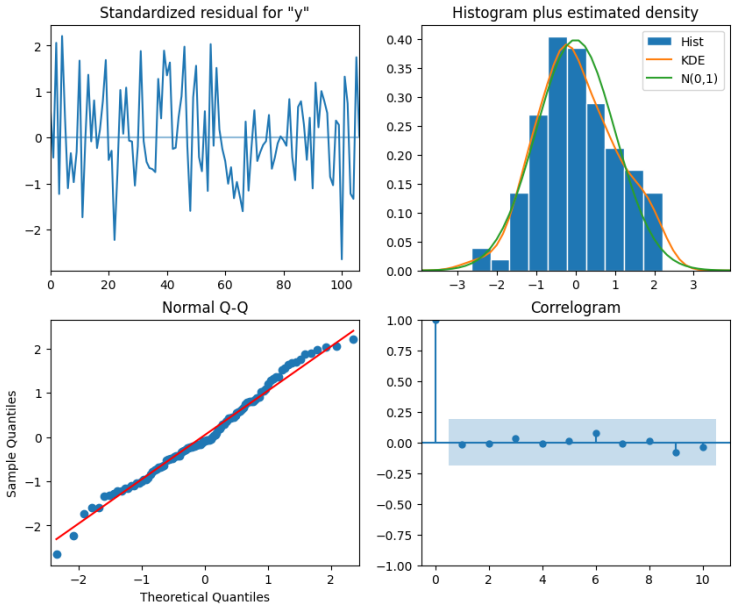

Figure 23: Residual plots for the ARIMA model

plot, the Q-Q plot shows that the points do lie in a linear fashion; and from the bottom-right plot, we see that the autocorrelation plot only shows a significant lag at lag 0, which means that there is no correlation between residuals. These plots again indicate that the residuals are similar to white noise, which means this SARIMA model is a good fit for the dataset.

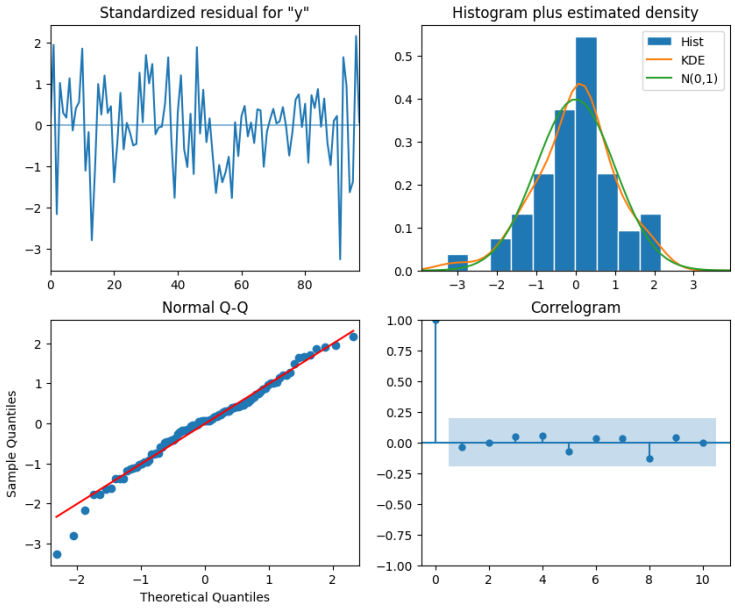

Figure 24: Residuals for the SARIMA model

Finally, for the ARIMAX model, we use the built-in function auto.arima in R the find the optima hyper-parameter. We set the range to be 0 to 10 for p, 0 to 5 for d, and 0 to 10 for q and then run the function. The optimal hyper-parameters were automatically chosen by the built-in function, which is p = 2, d = 0, and q = 2.

For Objective 2, the hyper-parameters were chosen based on an optimization function for training RMSE, because the autocorrelation plot gives unsatisfactory results. We choose ARIMA(1,1,1) as it best fits the actual values. A random hyper-parameter of p=2 was chosen for the VAR model due to the size of our data, which restrains the value of p to be 1 or 2 only, and 2 fits the data better.

5.2 Experimental Results

The experimental results will be presented in the following sections. Specifically, each section will present the results for each objective of this project. This includes the visualization of the predictions from the models, as well as the training and testing error for each model.

5.2.1 Objective 1

The first model that was applied for objective one was the LSTM model. Figure 25 and figure 26 show the model prediction on both the training set and the testing set.

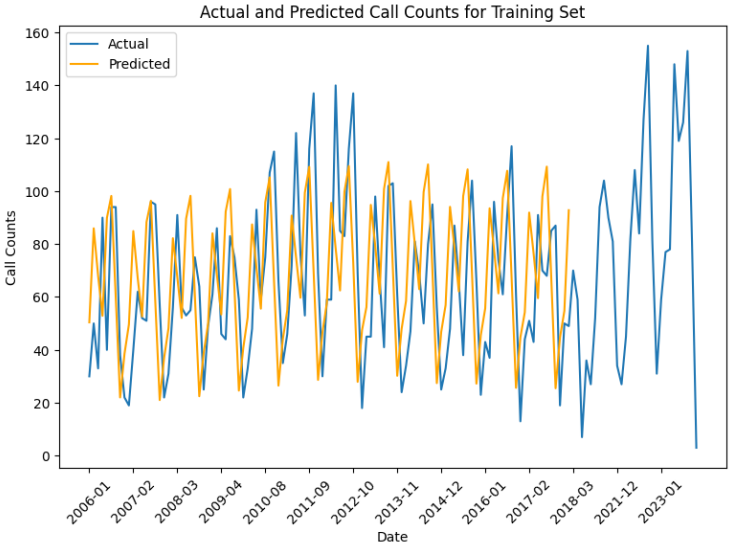

Figure 25: LSTM Training set

As we can see, the model is able to capture the trends from the data with moderate accuracy. However, even though the parameters selected for the model were the ones that produced the lowest testing error, there are clearly issues with this model for our dataset. First, our entire dataset has 140 data points in total. When we split 80% of the dataset into training set, we would have 112 data points in the training set, and that leaves 28 data points in the testing set. This is a relatively small number of data points to train a model with, especially with an LSTM model which was generally used for a larger dataset. Second, we proposed to have 128 neurons in both layers of the model. However, we have 112 data points in the training set, which means that the number of neurons actually exceeds the size of our training data. This could potentially pose a problem of overfitting for our model. Last but not least, the model we have produced a training RMSE of about 75.18 and a testing RMSE of about 28.96. The training error ended up exceeding the testing error, which could also be a sign of overfitting. Considering all reasons listed above, we decided not to move forward with the LSTM model.

On the other hand, the other models we used show some promising results. Figure 27 shows the prediction for both training set and testing set from both ARIMA and SARIMA, as well as the forecast for the next three years. As we can see, the predictions from both models are relatively similar, as they are able to predict the peaks and dips relatively well. In addition, both models seem

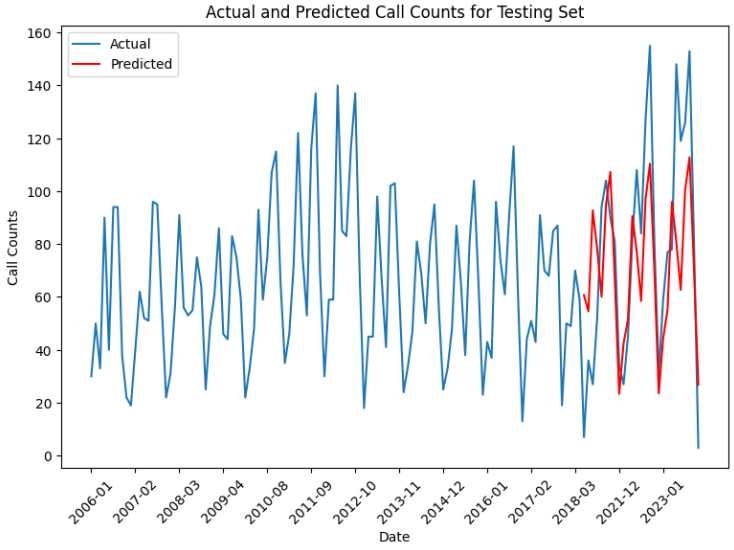

Figure 26: LSTM Testing set

to show an increasing trend in their forecasts for the future peaks, while the dips will stay relatively constant.

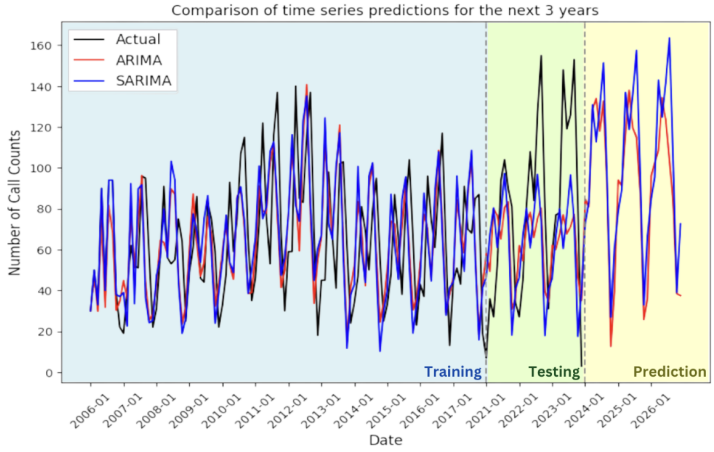

Figure 27: ARIMA and SARIMA

Figure 28 and Figure 29 show the results from the ARIMAX model. For the external variables, we choose the number of special events, the total enrollment, and the number of employees. Figure 28 shows the actual data as well as the model prediction for the training data (which is the first 80% of the entire data), and Figure 29 shows the actual data as well as the prediction for the testing data (which is the last 20% of the entire data).

Table 8 shows the training and testing error in terms of RMSE for each model. As we can see, although the time series data appears to have seasonality, the ARIMA model was able to perform better than the SARIMA model as it produced lower errors in both training and testing set. On the hand, we can see that the ARIMAX model has the lowest training and testing error out of all three models. One reason could be that the other variables in the dataset are closely correlated with call volume, and adding them as external variables increase accuracy in the prediction. For objective one, we conclude that ARIMAX is the most optimal model.

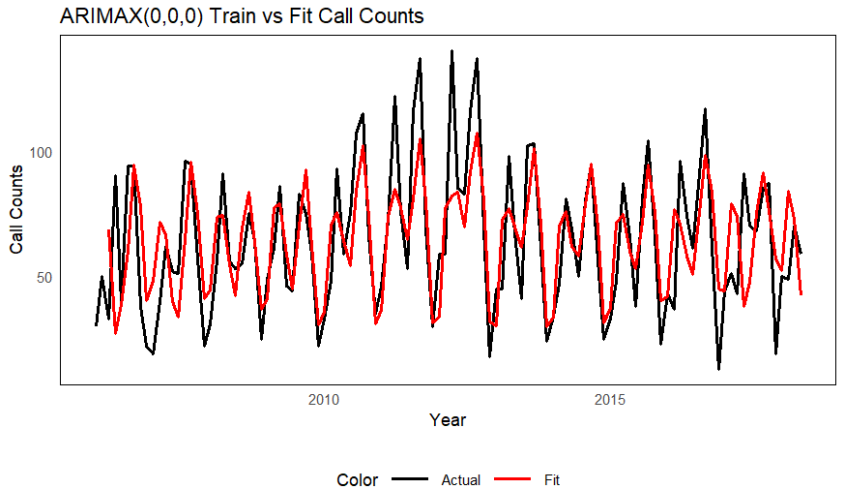

Figure 28: ARIMAX Training

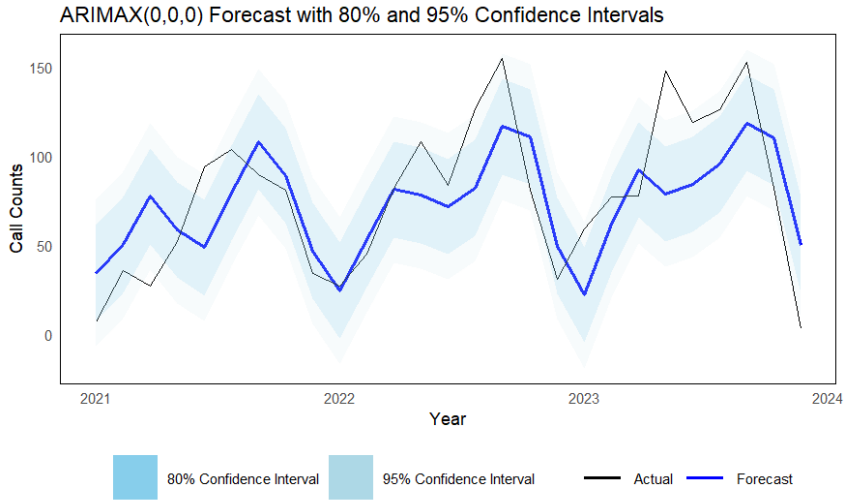

Figure 29: ARIMAX Prediction

Model	Training	Testing
ARIMA(10,1,0)	28.61	39.38
SARIMA(0,1,1)(1,1,1,9)	30.52	42.42
ARIMAX(2,0,2)	18.50	30.30

Table 8: Training and Testing error for each model

5.2.2 OBJECTIVE 2

The ARIMA, VAR, and polynomial regression model predictions for staff count over the next 3 years are displayed via graphs below in Figure 30, Figure 32, and Figure 32.

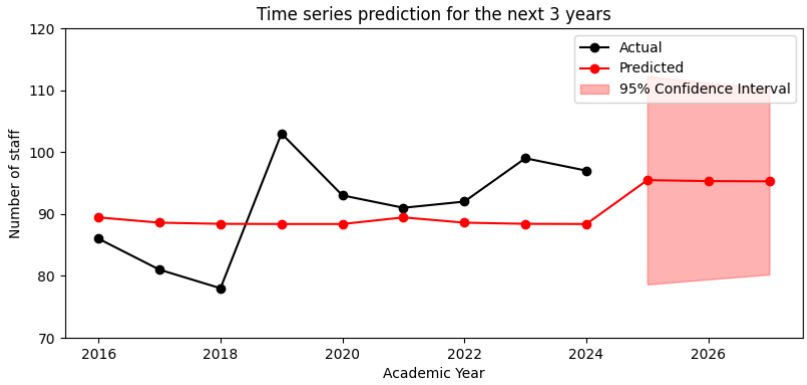

Figure 30: Predictions of total staff count for the next 3 years using ARIMA model.

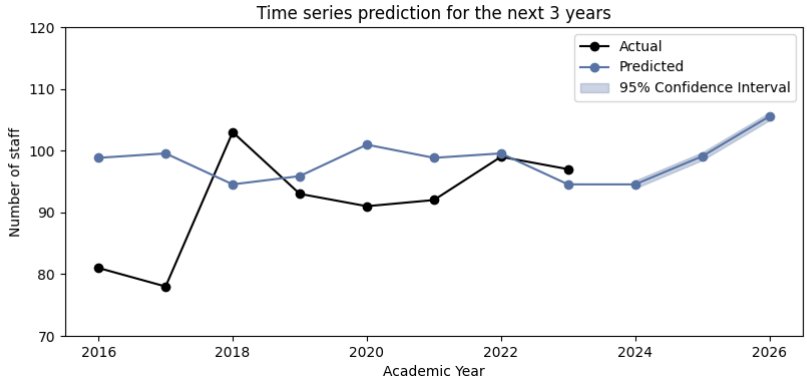

Figure 31: Predictions of total staff count for the next 3 years using VAR model.

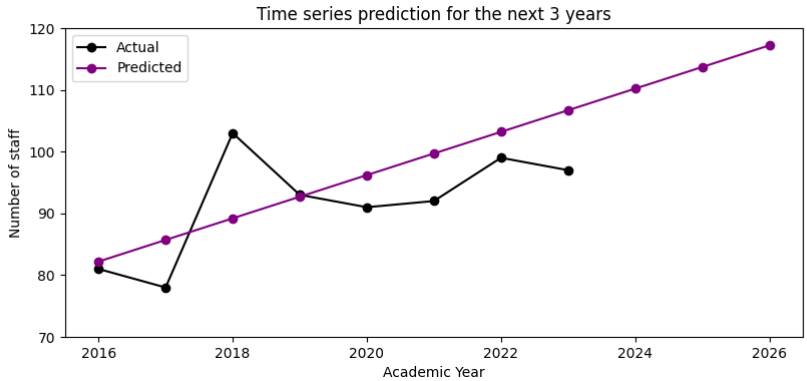

Figure 32: Predictions of total staff count for the next 3 years using polynomial regression.

Proceeding with the VAR model predictions, we expect an increase in total staff count over the next 3 years, showing the additional financial support needed from Rice for the development of REMS

services.The polynomial regression results suggest similar predictions, verifying the application of the time series models.

The predictions were made for both training and testing data. RMSE was calculated for both the training and the testing data as the error measure.

Table 9 shows the RMSE for the training and testing data for all 3 models:

Model	Training	Testing
ARIMA(1,1,1)	9.68	7.08
VAR(2)	13.07	4.21
Poly Reg(1)	7.46	7.55

Table 9: RMSE for each model

As we can see, all 3 models perform relatively well despite the data constraint. Their prediction results are summarized in Figure 33.

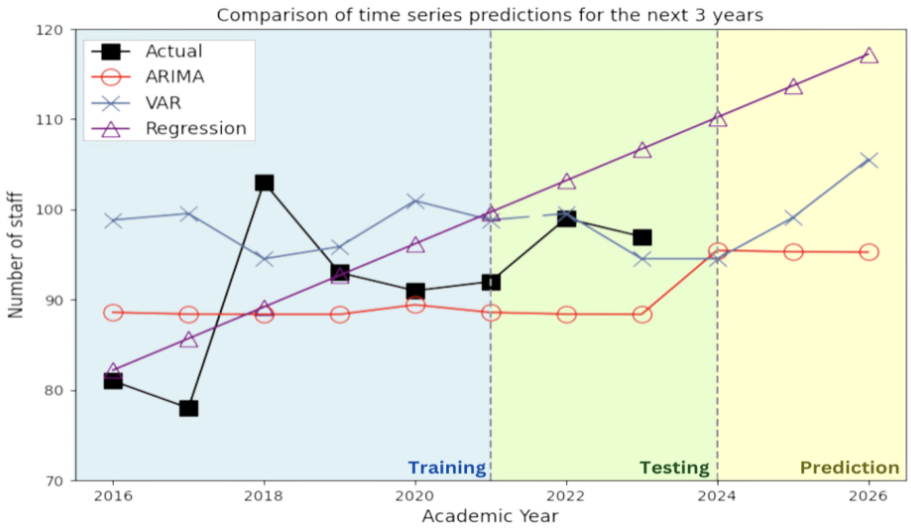

Figure 33: Comparison of predictions of total staff count for the next 3 years between 3 models.

For predicting the usage of the off-campus (OC) room, we noticed that the usage was rather seasonal, as seen in Figure 34. This means that the room was used less during certain months, like July, and more often in other months, like April. This means that instead of using ARIMA, we decided to use SARIMA to account for the seasonality of the data. The results of SARIMA show that there is expected to be a gradual increase in the usage of the OC room to about 800 monthly hours used. This value fluctuates based on the month we are in, increasing in the spring and fall and decreasing in the summer.

There was a peak in the data around 2020-2021 due to the fact that REMS was able to procure a second OC room. For those two years, REMS could use up to 48 hours of OC room a day, since there were two available rooms. However, once the pandemic began to recede and more people moved back on campus, they had to relinquish this second room. Currently, REMS has only one OC room. With an average monthly predicted usage of 800 hours, REMS has a good case to request a second room again; there are only about 720 hours in a month.

Model	Training	Testing
SARIMA(1,1,1),(1,0,2,12)	211	277

Table 10: Training and Testing error for SARIMA on OC Room

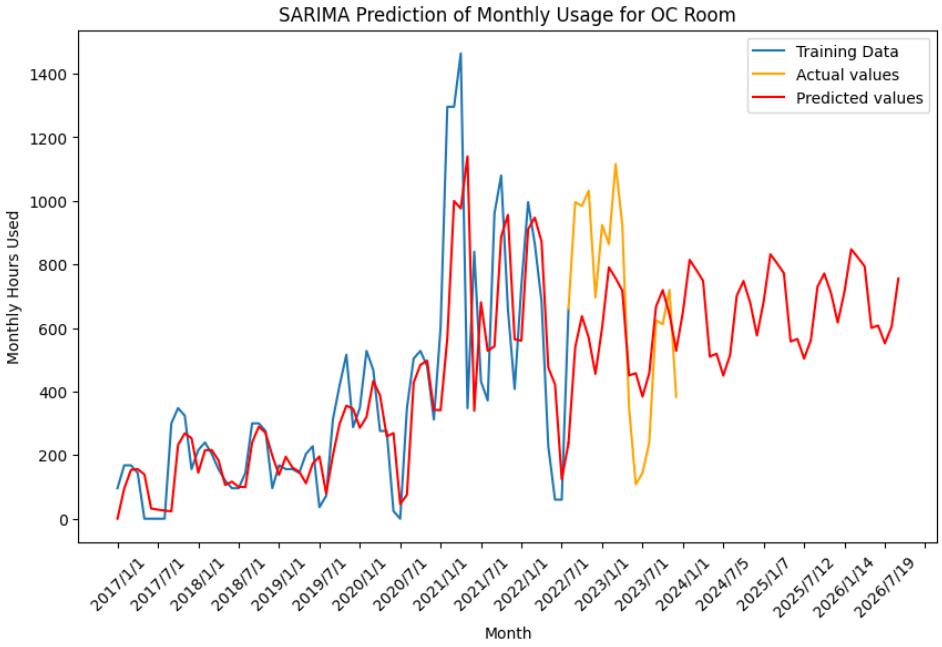

Figure 34: Training and Testing of the SARIMA model on the Off-campus room hours

5.2.3 Objective 3

The ARIMA and KNN model predictions for the five aspects of expenditures over the next 3 years are displayed via graphs below in Figure 35.

Table 11 shows the RMSE for the training set and the testing set for both models predicting the cost of vehicle maintenance.

Model	Training	Testing
KNN Regression	1016.20	2827.73
ARIMA(3,0,4)	1013.64	4524.95

Table 11: RMSE for each model predicting expenditure of vehicle maintenance

From Table 11, we find that KNN regression generated smaller testing error, which means for this dataset, KNN regression might be the better choice when predicting vehicle maintenance. However, since we have a training set containing only 5 data points and a testing set containing only 2 data points, KNN might not be the better model when the dataset gets larger in the future. The future forecast of the cost of vehicle maintenance for the future three years suggest just based on the data in the past 7 years, there is no need to increase the budgets a lot while having 700 more students enrolled. This might be reliable enough due to the small size of the dataset, which is a limitation on predictions of all five aspects of expenditures.

Table 12 shows the RMSE for the training set and the testing set for both models predicting the cost of IC housing.

From Table 12, we find that the two models generated similar testing error, which means for this dataset, there does not exist a better model when predicting vehicle maintenance. However, since we have a training set containing only 5 data points and a testing set containing only 2 data points, one of them might appear to be the better model when the dataset gets larger in the future. The future forecast of the cost of IC housing for the future three years suggest REMS might be able to decrease the budgets back to the level in the early years while having 700 more students enrolled.

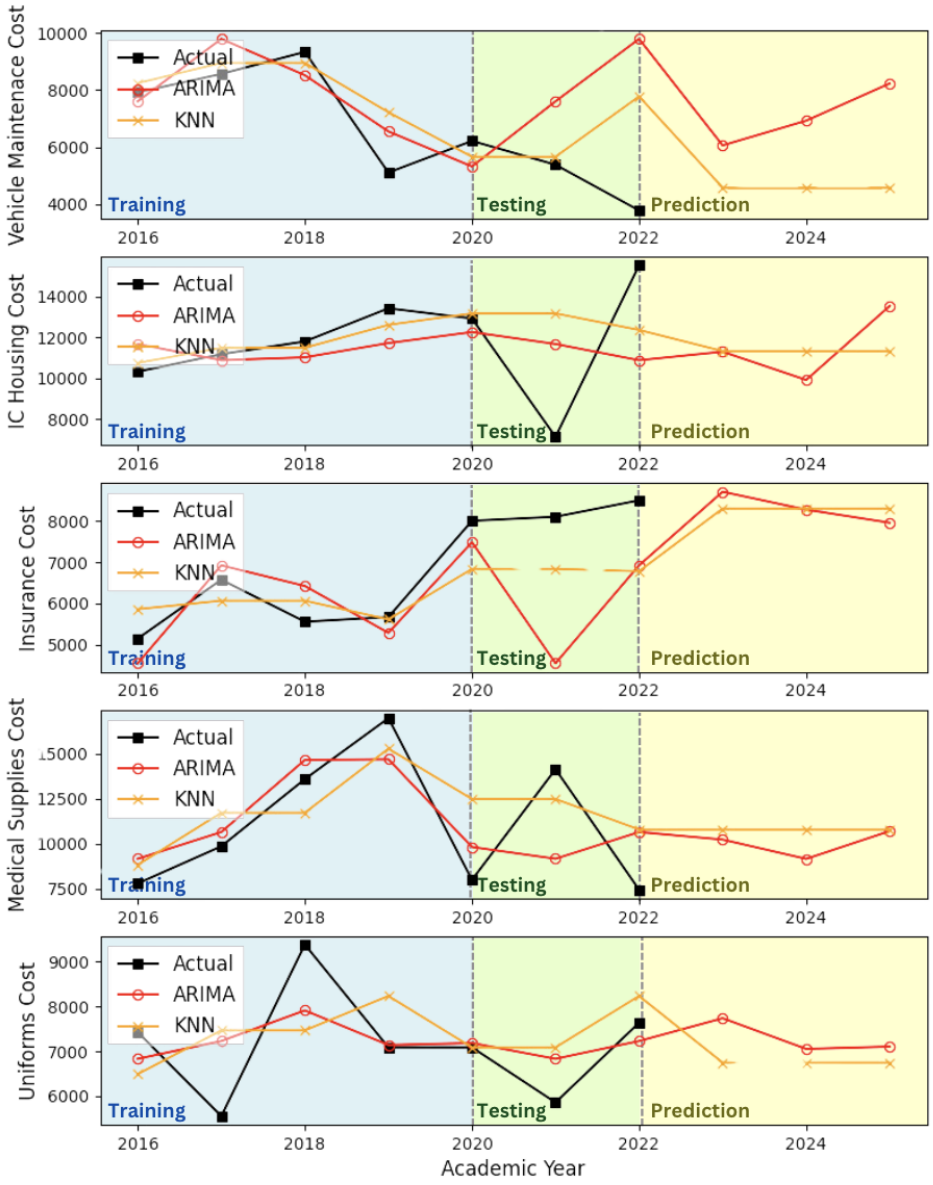

Figure 35: Predictions of the cost of vehicle maintenance, IC housing, insurance, medical supplies, and uniforms for the next 3 years using KNN regression and ARIMA

Model	Training	Testing
KNN Regression	467.62	4815.95
ARIMA(2,0,0)	1080.69	4587.14

Table 12: RMSE for each model predicting expenditure of IC housing

Table 13 shows the RMSE for the training set and the testing set for both models predicting the cost of insurance.

Model	Training	Testing
KNN Regression	689.63	1503.94
ARIMA(2,0,3)	571.95	2738.6

Table 13: RMSE for each model predicting expenditure of insurance

From Table 13, we find that the ARIMA model shows a better result on training set and KNN Regression model shows a better result on testing set, thus there does not exist a better model when predicting insurance now. However, due to the lower training error and the limitation of our dataset, ARIMA model may be the more suitable choice for predicting the insurance in the future.

By observing the result of insurance prediction in figure35, we can notice that the budget for insurance expenditure should be slightly increased in the next year. After the increase in the next year's budget, the department can maintain a stable insurance budget in the following years if there are no significant shifts in policy or coverage that would necessitate further increases.

Table 14 shows the RMSE for the training set and the testing set for both models predicting the cost of medical supplies.

Model	Training	Testing
KNN Regression	2495.56	2641.63
ARIMA(4,0,3)	1563.26	4188.39

Table 14: RMSE for each model predicting expenditure of medical supplies

From Table 14, we find that the KNN Regression model has a higher RMSE than ARIMA on the training set but a significantly lower RMSE on the testing set. Despite the ARIMA model's superior performance in the training set, the KNN Regression model exhibits greater stability between training and testing sets. This disparity suggests a potential overfitting of the ARIMA model to the training data, which undermines its generalizability to new datasets. Consequently, based on the criterion of model robustness, we believe that the KNN Regression model is the preferred predictive tool for estimating medical supplies.

Through Figure 35, we can see that REMS could plan for a slight increase in the budget for the upcoming year, with minimal changes in the subsequent years. Since the history data for medical supplies exhibited high volatility, while there is no need for a big change in the budget for medical supplies after the next year, it is still crucial to maintain some level of flexibility to respond to unexpected changes in the costs or needs of medical supplies.

Table 15 shows the RMSE for the training set and the testing set for both models predicting the cost of uniforms.

Model	Training	Testing
KNN Regression	1377.11	964.40
ARIMA(2,0,1)	1030.67	745.42

Table 15: RMSE for each model predicting expenditure of uniforms

From Table 15, we find that ARIMA model has a better performance in both training set and testing set, indicating it is the better model for this dataset. However, both models perform better on the testing set than on the training set, which is unusual in predictive modeling. Since we only have seven years data, we believe that this issue can be solved when we add more data to our models in the future.

For predicting the uniform in the future, we can see that in Figure 35, both models predict a decrease, a slightly reduction in the uniform budget could be justified in the future. The justification for this reduction is indeed sensible because uniforms can be recycled. By reusing uniforms and only replacing those that are worn out or damaged, the institution can reduce waste and save on costs.

6 CONCLUSIONS

6.1 IMPACT

In summary, our data science pipeline represents a novel and meticulously designed approach to address the objectives outlined by REMS, with a particular focus on enhancing emergency response planning at Rice University through predictive modeling of emergency call volumes. This pipeline stands out for its innovative integration of detailed data wrangling, exploration, and predictive modeling techniques.

The data wrangling process we developed is particularly innovative, converting yearly data into a monthly granularity and addressing missing values, which allowed for a more nuanced analysis. Notably, the decision to exclude summer months from our analysis reflects a tailored approach to meet the specific needs of REMS, focusing on academic year trends and thereby enhancing the relevance of our findings for our sponsor.

For objective one, we were able to use time series models to achieve desirable outcome. For objective two and three, because of the size of the dataset, there are limited approaches. However, our predictive modeling approach combined traditional time series forecasting methods with contemporary machine learning techniques. This blend is able to leverage the strengths of both statistical and machine learning models to capture complex temporal dynamics and the influence of external factors. Such an approach will be able to facilitate improved planning and resource allocation.

Moreover, the impact of our work extends beyond the technical realm, enabling REMS to make informed, data-driven decisions that optimize campus safety measures. The innovative aspects of our data science pipeline — ranging from advanced data wrangling to the strategic application of predictive models — demonstrate a forward-thinking approach to emergency preparedness. This not only contributes to a safer campus environment but also sets a precedent for the application of data science in university settings. In particular, from Objective 3, we helped REMS to know the potential increase/decrease of different kinds of expenditures, which allows them to plan for their budget in advance.

Looking ahead, the limitations and opportunities identified suggest pathways for further refinement of our methodology. The exploration of innovative data imputation techniques and the incorporation of seasonal variability analysis promise to enhance the robustness and accuracy of our predictive models. By continuing to adapt and refine these novel components, our data science pipeline will remain at the forefront of efforts to improve emergency response planning at Rice University, showcasing the transformative power of data-driven decision-making in enhancing campus safety.

6.2 FUTURE WORK

While our project presents a promising avenue for bolstering emergency response efforts at Rice University, it also highlights areas for future enhancement and exploration. One such area pertains to the challenge of missing data, notably in employee records and special event documentation. This limitation not only poses a risk of bias but may also compromise the precision of our predictive models. Advancing our collaboration with REMS to establish comprehensive data collection practices, along with employing innovative imputation methods, could significantly alleviate these concerns.

Additionally, our decision to exclude summer months from the analysis, while strategic, may limit our understanding of the full scope of seasonal dynamics affecting emergency call volumes. Future projects could benefit from integrating summer-specific data to uncover trends and insights pertinent to this period. Exploring distinct predictors for summer months and enhancing our models to more adeptly capture seasonal variations could yield a more holistic understanding of emergency call dynamics throughout the year.

For Objective 2 and Objective 3, the main limitation is that the datasets are too small for us to acquire reliable results. We believe as time goes on, the model will be more precise and we will then be able to identify which model performs better for each task in Objective 2 and Objective 3.

In conclusion, this project marks a significant stride towards leveraging data science in fostering a safer, more responsive campus environment at Rice University. The methodologies and insights garnered lay a solid foundation for ongoing and future efforts aimed at optimizing emergency medical services, with the ultimate goal of nurturing a secure and well-prepared campus community.

REFERENCES

Bimo Satrio Aji, Aniq Atiqi Rohmawati, et al. Forecasting number of covid-19 cases in indonesia with arima and arimax models. In *2021 9th international conference on information and communication technology (ICoICT)*, pp. 71–75. IEEE, 2021.

Fahad H Al-Qahtani and Sven F Crone. Multivariate k-nearest neighbour regression for time series data—a novel algorithm for forecasting uk electricity demand. In *The 2013 international joint conference on neural networks (IJCNN)*, pp. 1–8. IEEE, 2013.

Adebiyi A Ariyo, Adewumi O Adewumi, and Charles K Ayo. Stock price prediction using the arima model. In *2014 UKSim-AMSS 16th international conference on computer modelling and simulation*, pp. 106–112. IEEE, 2014.

Tao Ban, Ruibin Zhang, Shaoning Pang, Abdolhossein Sarrafzadeh, and Daisuke Inoue. Referential k nn regression for financial time series forecasting. In *Neural Information Processing: 20th International Conference, ICONIP 2013, Daegu, Korea, November 3-7, 2013. Proceedings, Part I 20*, pp. 601–608. Springer, 2013.

Peng Chen, Hongyong Yuan, and Xueming Shu. Forecasting crime using the arima model. In *2008 Fifth International Conference on Fuzzy Systems and Knowledge Discovery*, volume 5, pp. 627–630, 2008. doi: 10.1109/FSKD.2008.222.

Vinay Kumar Reddy Chimmula and Lei Zhang. Time series forecasting of covid-19 transmission in canada using lstm networks. 2020.

Jing Cong, Mengmeng Ren, Shuyang Xie, and Pingyu Wang. Predicting seasonal influenza based on sarima model, in mainland china from 2005 to 2018. *International journal of environmental research and public health*, 16(23):4760, 2019.

Nemanja Deretić, Dragan Stanimirović, Mohammed Al Awadh, Nikola Vujanović, and Aleksandar Djukić. Sarima modelling approach for forecasting of traffic accidents. *Sustainability*, 14(8): 4403, 2022.

Jamal Fattah, Latifa Ezzine, Zineb Aman, Haj El Moussami, and Abdeslam Lachhab. Forecasting of demand using arima model. *International Journal of Engineering Business Management*, 10: 1847979018808673, 2018.

Fatemeh Gholamzadeh and Sara Bourbour. Air pollution forecasting for tehran city using vector auto regression. In *2020 6th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS)*, pp. 1–5. IEEE, 2020.

Farid Kadri, Fouzi Harrou, Sondès Chaabane, and Christian Tahon. Time series modelling and forecasting of emergency department overcrowding. 2014.

Fazle Karim, Somshubra Majumdar, Houshang Darabi, and Samuel Harford. Multivariate lstm-fcns for time series classification. 2019.

Firdos Khan, Alia Saeed, and Shaukat Ali. Modelling and forecasting of new cases, deaths and recover cases of covid-19 by using vector autoregressive model in pakistan. *Chaos, solitons & fractals*, 140:110189, 2020.

Edson Zangiacomi Martinez, Elisângela Aparecida Soares da Silva, and Amaury Lelis Dal Fabbro. A sarima forecasting model to predict the number of cases of dengue in campinas, state of são paulo, brazil. *Revista da Sociedade Brasileira de Medicina Tropical*, 44:436–440, 2011.

Rangsan Nochai and Titida Nochai. Arima model for forecasting oil palm price. In *Proceedings of the 2nd IMT-GT Regional Conference on Mathematics, Statistics and applications*, pp. 13–15. Academia Penang, 2006.

Marco Peixeiro. *Time series forecasting in python*. Simon and Schuster, 2022.

Sima Siami-Namini, Neda Tavakoli, and Akbar Siami Namin. The performance of lstm and bilstm in forecasting time series. In *2019 IEEE International Conference on Big Data (Big Data)*, pp. 3285–3292, 2019. doi: 10.1109/BigData47090.2019.9005997.

Kinley Wangdi, Pratap Singhasivanon, Tassanee Silawan, Saranath Lawpoolsri, Nicholas J White, and Jaranit Kaewkungwal. Development of temporal modelling for forecasting and prediction of malaria infections using time-series and arimax analyses: a case study in endemic districts of bhutan. *Malaria Journal*, 9:1–9, 2010.

Sun Xiumei, Zhou Min, and Zhang Ming. Empirical study on the relationship between economic growth and carbon emissions in resource-dependent cities based on vector autoregression model. *Energy Procedia*, 5:2461–2467, 2011.