BUS ROUTE DEMAND PREDICTION WITH DEEP LEARNING

[1]Stevanus Alditian Lai, [2]Yonathan Purbo Santosa
[1,2]Program Studi Teknik Informatika Fakultas Ilmu Komputer,
Universitas Katolik Soegijapranata
[2]yonathansantosa@unika.ac.id

ABSTRACT

bus companies currently have several obstacles in providing their fleets from one city to another because of the highly dynamic demand from passengers, bus companies must be able to analyze which routes will have a lot of demand so that bus companies can provide more fleets on the routes that will have high demand. Deep learning method is relatively new for bus company to predict the bus route demand, this study is try to create and implement LSTM Autoencoder-Bi-LSTM Hybrid Models and Bi-LSTM to forecast bus route demand to support the decision making process in order to optimize bus fleet deployment each route. The results shows that LSTM Auto-encoder-Bi-LSTM Hybrid Models and Bi-LSTM models doesn't differ very much.

Keywords: Autoencoders, deep learning, LSTM, Bi-LSTM, Timeseries

INTRODUCTION

The transportations become a human need to support mobility, there are 3 types of Transportation, namely land transportation, sea transportation, and air transportation. Transportation that is often found and used is land transportation because the price is relatively cheap and easy to reach by the public. The means of transportation are divided into 2 parts, namely public transportation and private transportation. Public transportation until now has become one of the most efficient means of transportation because it is able to carry many passengers at once and is quite practical. Public transportation that is often used by people to go somewhere is a bus, especially if you want to go out of town, the bus is a good choice. The bus is one of the modes of land transportation that can accommodate many people at once. However, bus companies currently have several obstacles in providing their fleets from one city to another because of the highly dynamic demand from passengers, bus companies must be able to analyze which routes will have a lot of demand so that bus companies can provide more fleets on the routes that will have high demand. Unfortunately, the bus company is currently still unable to predict which routes will be in high demand, at this time the bus company can only guess. Currently, to overcome this, the bus company has collected data which will later be analyzed.

Time series is a collection of data where the data are indexed by date or time sequentially. Multi variate time series is a type of timeseries data but has many data variables. Bus route ticket sales when collected over time and indexed by date and time it becomes time series data. There are many methods and algorithms that can be used to analyze the time series data such as artificial intelligence, machine learning, data mining to deep learning. Machine learning is a study that try to make the computer learns from huge data set to solve specific problems that provided. Until now machine learning model has solved many problems and contribute to the development of the

world. Now day's machine learning is used to translate a language to another language, predicting time series data, classifying problems, and clustering data from big data. Big data is a stream of data that have extremely high velocity, large variety, and large volume.

Time-series forecasting is highly depending on historical data. Besides that, time-series data have many variables that can affect the results of the forecasting, therefore a method before forecasting is probably needed to extract features from the time series data. There will be some variable that will impact much bigger than other variable for forecasting bus route demand, and from one variable we can get more information that should impact the forecasting. The hypothesis is the data of event each date will impact the most and from date many new information can be extracted out from it, like semester break (students are more likely to go home to their hometown), or the Christmas and new year, for example. Without a doubt, LSTM does have good performance in handling continuous data, but LSTM itself probably cannot handle data that have very large dimensions like this multivariate time-series, therefore there is a model architecture that can help LSTM to make predictions called autoencoder. Autoencoder is an architecture that has 2 stages, namely the compression stage and the decompress stage, this is done so that the model tries to make a copy of the original data to extract features on very large dimensions. In this research bi-LSTM-Autoencoder and bi-LSTM models will be compared to see the effect of autoencoders in bus route demand forecasting problems. Since LSTM can have a memory of past events, LSTM models could use to solve the forecasting problem.

LITERATURE REVIEW

Time series data is a collection of data which is indexed by timestamp. Time series (time series) forecasting is a process to predict the future by analyzing the historical data. The assumption of time series forecasting is that the information will repeat itself in the future[1]. Multivariate Time Series (MTS) data is a type of time series data that typically very high dimensional [2]. according to [1] Plenty of research has been done in time series analysis to accomplish multiple objectives. Even in today's machine learning and deep learning era, time series forecasting plays a crucial role to make important business decisions. The models that are frequently used are Convolution Neural Networks, Long Short-Term Memory and GRU models, each of them have their own uniqueness.

A comparative study of the frequently used models (Convolution Neural Networks, Long Short-Term Memory and GRU) to predict time series data has been done in [3], [4]. K. Nazmoon, T. Tahmid, A. Rafi, and M. Ehsanul, [3] in their research, the author try to forecast covid case in brazil, UK and Russia also compare the performance of Convolution Neural Networks and RNN base models for the problems. Convolution Neural Networks model have ability to recognize pattern so it will suit for time series problem because like mentioned in [1] The assumption of time series forecasting is that the information will repeat itself in the future. Long Short Term Memory in other hand can handle large dataset especially to handle data with seasonality. The result of K. Nazmoon, T. Tahmid, A. Rafi, and M. Ehsanul research is for covid case Convolution Neural

Networks algorithm perform better than other method. This probably because covid case data doesn't have seasonality in its so RNN approach do not perform well enough in this problems.

Other comparative study to compare Convolution Neural Networks models and RNN models is done by [4] by comparing Convolution Neural Networks, Long Short Term Memory, GRU and MLP to solve univariate time series and multivariate time series. the goal is find the optimal deep learning models that best suit for univariate time series and Multivariate time series. The author found that Long Short Term Memory works the best for a highly dynamic dataset like opening prices of a stock market and GRU works well for a seasonally repeating dataset like temperature [4]. and for the multivariate time series forecasting Long Short Term Memory gives the least RMSE and MLP has the highest RMSE for a comparatively stable dataset [4], which lower RMSE value means more better prediction performance. From this research the author get that Long Short Term Memory and GRU is more efficient and suit for time series forecasting than other models.

The research from [3], [4] have different results, from [3] research where the researcher found that the Convolution Neural Networks models are works better than the other models and from [4] where the researcher found that LSTM and GRU models do better job for forecasting time series value. But from [3] and [4] research the difference between the result probably because the number of data that each of them use where [3] use Brazil, Russia, and the United Kingdom data of 291 days are used for training, and data of 33 days are used for validation and for [4] the author use 1258 daily values from 03-01-2012 to 31-12-2016 for the google stock price and for the occupation datasets the author use roughly 20560 observations. That means the LSTM and GRU models work better if more data is fed then Convolution Neural Networks will work better than LSTM and GRU when data is limited. This phenomenon occurs probably because of Convolution Neural Networks have ability to find pattern from data while LSTM and GRU has its own memory cells to memorize the dependents of historical data to future data so LSTM and GRU can overcome Convolution Neural Networks models if big data is fed into its.

Long Short Term Memory model has been used in many time series problem, Long Short Term Memory also has been used by [5]. in this research I. Sülo, Ş. R. Keskin, G. Doğan, and T. Brown use Long Short Term Memory to achieve smart and energy efficient building by predicting building energy consumption. Energy efficiency can be defined as utilizing the amount of energy consumed with the least waste of resources or effort, intelligent buildings stand out as a process that is carried out in a controlled manner within the building systems, in order to maximize energy efficiency [5]. the author do experiment to this data, firstly the author try to get Prediction of the Energy Consumption Values as One- Dimensional Data and the result is not sufficient than the author try to give temperature variable to find Effect of Temperature on Energy Consumption and the result for the energy consumption prediction is the predictions become much more accurate, then the author add humidity variable from datasets and the prediction performance is more better than just 1 variable, lastly the author add occupancy variable however since the occupancy variable have missing values then the prediction accuracy was drop[5]. From this research we get that missing value is a big problem for time series forecasting it must be handled first before the data are

used. There are some research that try to handle missing data like the research [6] in here the author propose new method called GRU-D which similar like GRU models but this model isa a trainable decay models in which a decay mechanism is designed for the input variables and the hidden states to capture the aforementioned properties [6]. they have assumption that missing values tend to be close to some default value if its last observation. And the result of proposed GRU-D models is promising.

Other problems for multivariate time series data are due its high dimension, which can be a pit fall for deep learning models. The high dimensionality probably need to handled first before data is fed into models. There are several techniques that used to handle this high dimension data for example the principal component analysis (PCA). PCA has been used in many research. Like the research that has been done by K. Yang and C. Shahabi in [2]. [2] in this research the author try to adopt kernel PCA to reduce the dimension out of MTS data sets the author called this method the Generalized Principal Component Analysis (GPCA). The result of this study is GPCA is worse than the Kernel PCA technique.

While [2] said that vectorization is essential Ouyang, Kewei, Hou, Yi, Zhou, Shilin, Zhang, Ye [7] said the case for time series is global translation invariance and The scale and stages changes due to temporal distortion that bring significant challenges to TSC. In [7] in this research Ouyang, Kewei, Hou, Yi, Zhou, Shilin, Zhang, Ye propose 3 architecture EM-Convolution Neural Networks architectures including elastic matching FCN (EM-FCN), elastic matching ResNet (EM-ResNet) and elastic matching Inception (EM-Inception) Inspired by the elastic matching in dynamic time warping (DTW). Because the conventional Convolution Neural Networks can't handle the temporal distortion problem. Dynamic time warping is a point-to-point matching method to measure the similarity between two different time series. In general, DTW allows a time series to be "stretched" or "compressed" to provide a better match with another time series [7]. The result of [7] research found that the performance of the EM-FCN, EM-ResNet, and EM-Inception are better than the FCN, ResNet, and Inception, respectively. Ouyang, Kewei, Hou, Yi, Zhou, Shilin, Zhang, Ye [7] noted that the EM-Convolution Neural Networks is not better than the corresponding Convolution Neural Networks in all the datasets and the base architecture is important to the performance. It is more helpful to combine the elastic matching mechanism with the Convolution Neural Networks in the "motion" datasets.

PCA is in [8] research the author is trying to predict stock price using PCA-LSTM model. the model uses PCA to extract the main feature out of a number of technical indicators that affect stock prices, to reduce the dimensionality, reduce training time, and improve model performance. After extracted, the component then fed into Long Short Term Memory model to predict the stock price. In Wen, Yulian, Lin, Peiguang, Nie, Xiushan [8] found that PCA-LSTM hybrid can predict time series data better than conventional Long Short Term Memory thanks to PCA ability to extract and reduce the dimensional. Other method for extracting the feature is done by Y. Yuan et al. In [9]. in Y. Yuan et al. Research the author try to use RNN-DNN base to extract feature to help

predict travel time of bus. But this method has not been tested enough and need to be more improved since the author need face missing value first.

Besides deep learnig, machine learning approach can be used to predict time series problem. The most frequently used machine learning aproch to predict time series is the SVM models. SVM can become a handy alternative. Research that tries to compare SVM and RNN is research that done by L. Badal and S. Franzén, in [10]. in this research the author tries to compare the predicting acuraccy of RNN and SVM for forcasting energy price. The result is quiet supprising that the SVM models achive better accuracy than deep learning (RNN) models.

From all the research that reviewed, they have used many kinds of deep learning and machine learning algorithm for time series problems. However, from 10 research reviewed there is no research that use deep learning models to solve time-series problems for bus route forcasting. This research is about to implement deep learning models to solve the bus route forcasting problems.

RESERCH METHODOLOGY

Data Anlaysis and Preprocessing

This is the very first step when making deep learning models, because this step is highly crucial and will impact the deep learning model performance and result. The data from this stage will be fed into the model to make predictions, so it is important that the data has been processed correctly so that the model can learn and make better predictions. To perform data analysis and processing, there are several stages to be carried out, namely: data selection, data visualization and variable analysis and data preprocessing.

Data Selection and Variable Analysis

At this step, the variables will be analyzed in order to select the variables that relevant for the task and will affect the prediction of bus routes and drop data variable that is not relevant for this task, as already mentioned in the background, data may contain valuable information that can help forecast. Therefore, before continuing, the data needs to be analyzed first to extract hidden information.

Data Selection

Data selection is a step to selecting and dropping the used and unused data given from the source, this must be done because not all data variable is related to the problem that want to be solved, so the variable must be dropped, The data variable given from the company is as shown below :

1. scheduleDate : The passenger's departure date time is shown in this variable.
2. Start: The origin of the passenger's departure is shown by this variable.
3. Destination : The passenger's destination is shown in this data variable.
4. Qty : total of passeger per transaction
5. Name : The name of the bus route
6. className : the class of the bus(ECONOMY, AC, etc)

from all data variable above the used variable only scheduleDate, start, Destinations and Qty.

Data Variable analysis

Indeed, the data variables selected is quite small but with a deeper analysis, some variables have hidden information contained in it which is able to help deep learning training from the data so that predictions from the model can be obtained better.

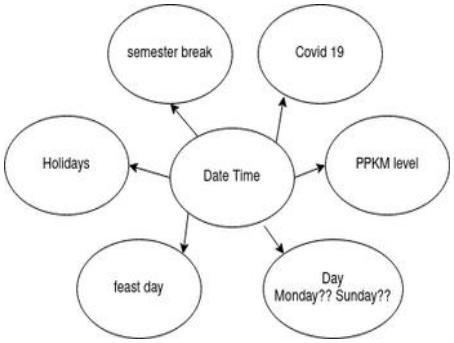

Figure 1. Date Time variable analysis

From figure 1, it is explained that date time has a lot of information that can be used as new variables and these new variables can also affect bus demand. For example, we know that bus demand will rise when holidays are coming.

Data Visualization

Data Heatmap provided will be analyzed more and data visualization will be created in form of heatmap and visualization of data distributions to achieve clear visualization. Data visualization is done to get clear image so further analysis and action on the data can be carried out.

Heatmap Visualization

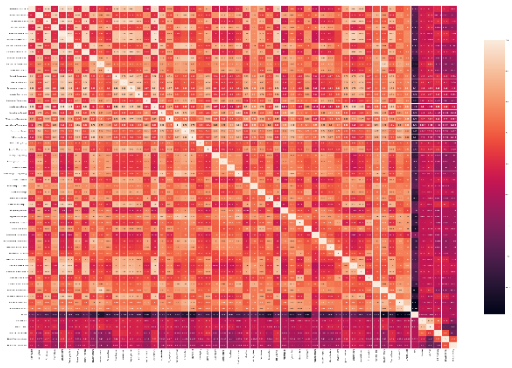

Figure 2. Heatmap

With help of seaborn library to create heatmap to achieve clear visualization about how each variable correlates with each other and how strong the correlation is, so it can be ensured that the

data can be predicted. Heatmap is a visualization method that shows how each data variable correlate with each other. Data variables can have correlation with other data either positively correlate or negatively correlate with each other or even the data will not have correlation at all. Heatmap visualization is important in time series task since the time series task will have many variables and each variable must have correlation with one another, if not then the time series data could not be predicted further.

Histogram Visualization

In addition to the heatmap, a histogram illustration will also be carried out. Histogram here is used to show how the data distribution is, so if the data distribution is not normal then further step could be done besides that from histogram get a clear visualization of the Gaussian distribution of the data in order to get a clear picture of what actions should be taken to make the data more have more gaussian like distribution.

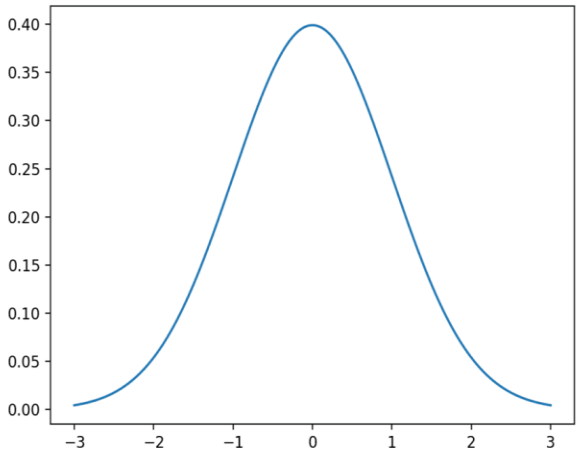

Figure 3. Gaussian distribution

Split Data

The data that was previously analyzed will be entered and split in this stage. The data will usualy divided into two parts, the dependent variable and the independent variable. Dependent variable is a data variable that depend on the independent variable, and indepenent variable is data that dosent depand with other data. with the dependent variable being the bus route that will be predicted and the independent variable is the categorical data. In order to classify the data per record in time-series, and then the data will be divided into two parts, 80% training set and 20% test set.

Feature Scaling

The data that have been splited then must be scaled first. Its unkown for what feature scaling is the best for machine learining and deep learning models. But feature scaling is important to performed for the data so deep learning or machine learning models could perform better job.

There are many method to do feature scaling, normalizaation, standarization, power transformer an so on. Normalization is a method to normalize the data and scaled into bettween 0 an 1 value. Equation below is showing how the normalization is calculated.

$$X_{norm} = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (1)$$

Standarization is a method to transform the data with its standard deviation. Figure below is the formula of standarization method, it seen that the x value is substracted with the mean times x then divided by the standard deviation as described in equation 2. After standarization performed the data mean should be near 1.

$$X_{norm} = \frac{x - \text{mean}(x)}{\text{std(x)}} \quad (2)$$

It must be noted that feature scaling is must be done after the data is split because if feature scaling is done before then the split data could not be inverted back into its normal form. is a data variable that depend on the independent variable, and independent variable is data that doesn't depend on other data. with the dependent variable being the bus route that will be predicted, and the independent variable is the categorical data. To classify the data per record in time-series, and then the data will be divided into two parts, 80% training set and 20% test set.

Creating Models

The models that are frequently used are Convolution Neural Networks, Long Short-Term Memory and GRU models, each of them have their own uniqueness [1]. This research will use LSTM-Autoencoders-Bi-LSTM and Bi-LSTM models to forecast the bus route demand then Autoencoder-Bi-LSTM and Bi-LSTM Models performance will be compared to find autoencoders architecture effect on time series forecasting problem. 60% of data will become the training set for the deep learning to learn, and the models will validate the gained weight with the help of validation sets then the last to calculate the performance will be explained in next part.

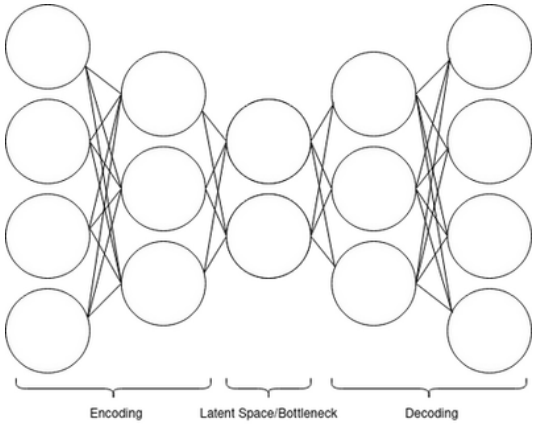

Figure 4. Autoencoder architecture

LSTM Autoencoder-Bi-LSTM Hybrid Models

The autoencoders will have 2 stages the first stage is the encoding stage, in encoding step the large data variable sequence will be compressed into bottleneck/latent space then after encoding the data then will be decompressed again and try to decode the latent space variable back to its normal dimension when trying to reconstruct the data variable back to its original form to hopefully models got new information that will help the models to learn more from the data.

Bi-LSTM Model

As mentioned before Bi-LSTM is an artificial neural network layer that based on LSTM but learns the data Bidirectionally. This model will use Bi-LSTM models to create a comparison to find the effect of autoencoders in time series problem. For this model the data will be fed directly into the models and Bi-LSTM will directly calculate and provide the predictions per sequences.

Models Evaluation and Comparison

The performance of the system will be evaluated in this part. Since the forecasting problem would have high error rate the metrics used will be RMSE because RMSE will root the given error first so RMSE will give more weight to larger error and for the training loss function is MSE. RMSE is calculating the root-mean-square-error of the predicted and the real value. RMSE metrics are calculated in equation 3 and 4.

$$RMSE = \sqrt{\frac{1}{n}\sum_{j=1}^{n}\left(y_j - \hat{y}_j\right)^2} \tag{3}$$

$$MSE = \frac{1}{n}\sum_{j=1}^{n}\left(y_j - \hat{y}_j\right)^2 \tag{4}$$

2 models will be made namly the LSTM Autoencoder-Bi-LSTM Hybrid Models and Bi-LSTM models, so that comparisons can be made by looking at the RMSE results obtained. After training finished the models loss and metrics history during training epoch will be saved for each model's then compared to find the effect of autoencoders in time series forecasting task.

ANALYSYS AND DESIGN

Data Preprocessing

Preprocessing the raw data is the first step in building deep learning models. This step is crucial when creating deep learning models because if this step goes wrong, then the other part will also go wrong as well. Data preprocecessing step is part where the raw data proccessed before inputed into deep learning models. Pre processing part wiil be splited into 3 parts:

1 Data Selection and Variable Analysis
2 Feature Extraction

3 Split Data
4 Data Analysis and feature scaling

Data Selection and Variable Analysis

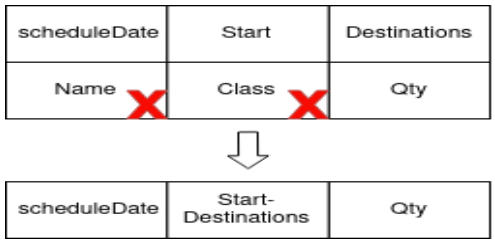

Figure 5. Data Selection

The first step is to selecting the data variable and drop variable that will not used. As shown in Figure 5 in this problem Name and Class wariable will be dropped since rus route deman will rise when something triggering the people to use busses not what the bus company give facility so Class variable doesent effect on bus route demand, and Name variable is also droped and start, destination vaariable will combined into 1 variable. The start and destination data are initially the names of the departure and destination points for passengers and one city may have more than 1 point so that the values for start and destination are changed to the name of the city of the departure and destination points.

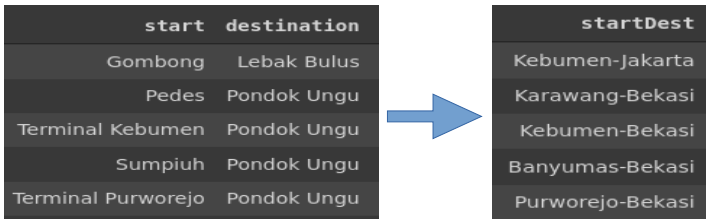

Figure 6. : Start-destination transformation

As shown in Figure 6, the start and destination variable are transformed to its city name then combined, this method will reduce the amount of start-destinations count without loosing information like in Figure 6 first data, the “gembong” is inside the city of Kebumen and lebak bulus is inside city of Jakarta. Then after this step the data will be pivoted so each unique startDest will become a new variable and the value of each variable is the quantity of passenger on a certain date that are illustrated in Figure 7.

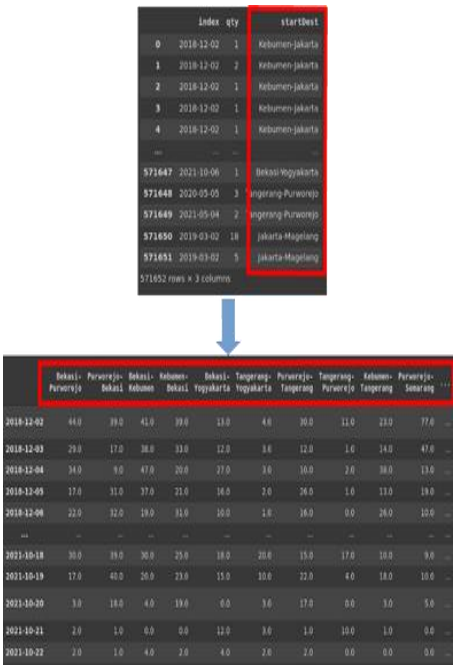

Figure 7. Pivoting each unique startDest

From the Figure 7 it shows that the data have new structure with the variable now is the startDest and the value for each unique startDest is presenting its own passenger quantity at certain date so the goal of this research that to predict each route on certain date can achieved.

Feature Extraction

After Data Selection and Variable Analysis is performed the next part is to find supporting variable that needed to get better model performance result later. Although the data of route itself when fed into deep learning is possible, but we need to figure how to make deep learning models can perform better job on this, so we should think what thing should affect the bus route demand. As mentioned last chapter we know that many features can be extracted from date variable so we should analyze more about what feature can be extracted from a date time variable.

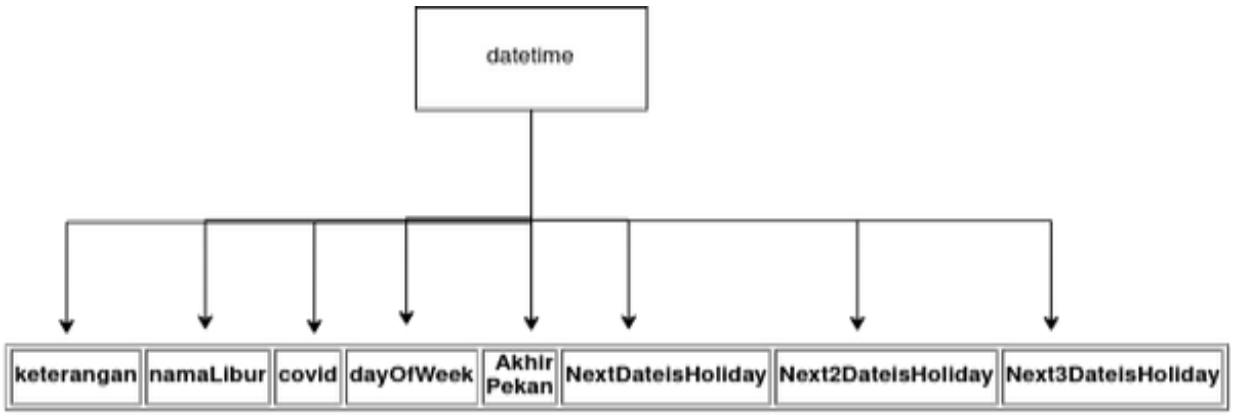

Figure 8. date time feature extraction

As shown in Figure 8 from date time data many variables can be extracted. The variable extracted is “keterangan”, “namaLibur”, “covid”, “dayOfWeek”, “Akhir Pekan”, “nextDateisHoliday”, “Next2DateisHoliday”, “Next3DateisHoliday” is obtained. Keterangan is variable that represent about certain date is holiday, normal day, and weekend, if holiday then keterangan value is either be “liburNasional” or “cutiBersama” depends on the namaLibur variable. Namalibur variable is representing the name of holiday, like “Libur Natal”, “Libur lebaran” etc. Then from date time we should know the certain date is in COVID-19 pandemic or not because since COVID-19 pandemics the bus demand is hugely drop. Then day of week is represented name of the day, for example Sunday, Monday, Tuesday, etc. nextDateisHoliday is represents that in certain date plus one day is a holiday or not, this should impact bus route demand because people tend to go to their hometown or taking a vacation and probably use bus as their transportation when the next day is holiday, and for the next2Dateisholiday is like the nextDateisHoliday but this variable represents that in certain date plus two day is either holiday or not, and the last one next3DateisHolidat is a same as nextDateisHoliday but his variable represents that in certain date plus three day is either holiday or not.

Split Data

The data that was previously analyzed will be entered and split in this stage. The data will be divided into two parts, the bus route variable and the categorical variable that extracted from the date time variable. The bus route variable is containing the data of the bus route passenger sum on certain date. The data then will be divided into two parts, 80% training set and 20% test set.

Data Analysis and feature scaling

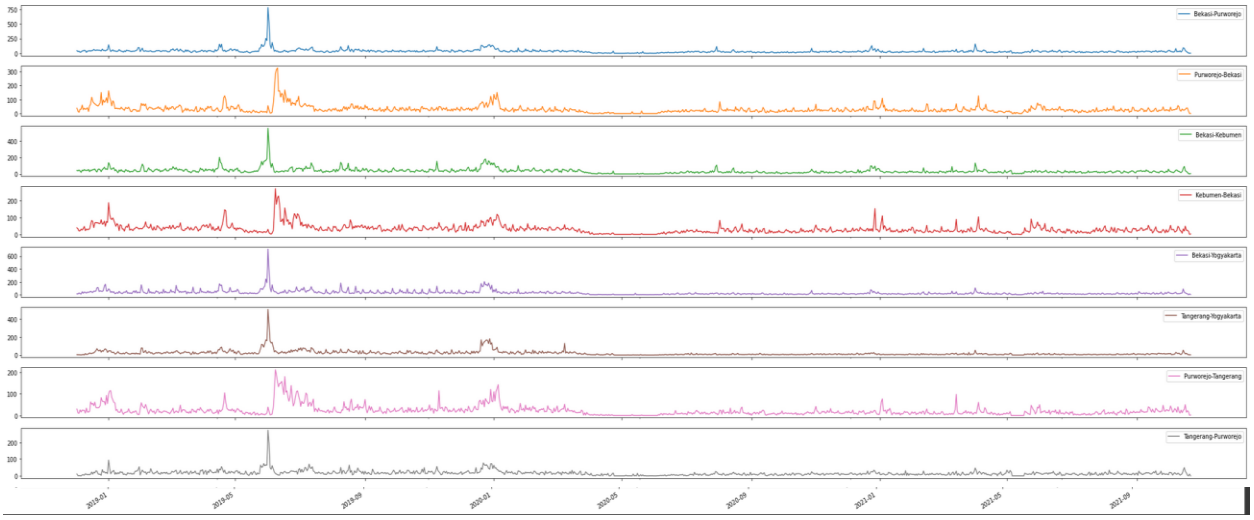

Figure 9. data visualization

This part is to analyze the data that has gone through the previous two stages, this step will use seaborn to visualize the corelation of each variable and data distribution will be visualized with histogram then feature scaling method will be performed. Before that the author visualize the data

first to get track of missing values. Figure below is the visualization of some of the route data. From Figure 9 we can see that the data there was a very significant decrease in the number of passengers in Marc and that time is the first time COVID-19 outbrakes in Indonesia. Besides that, there are no evidence of missing values in this data.

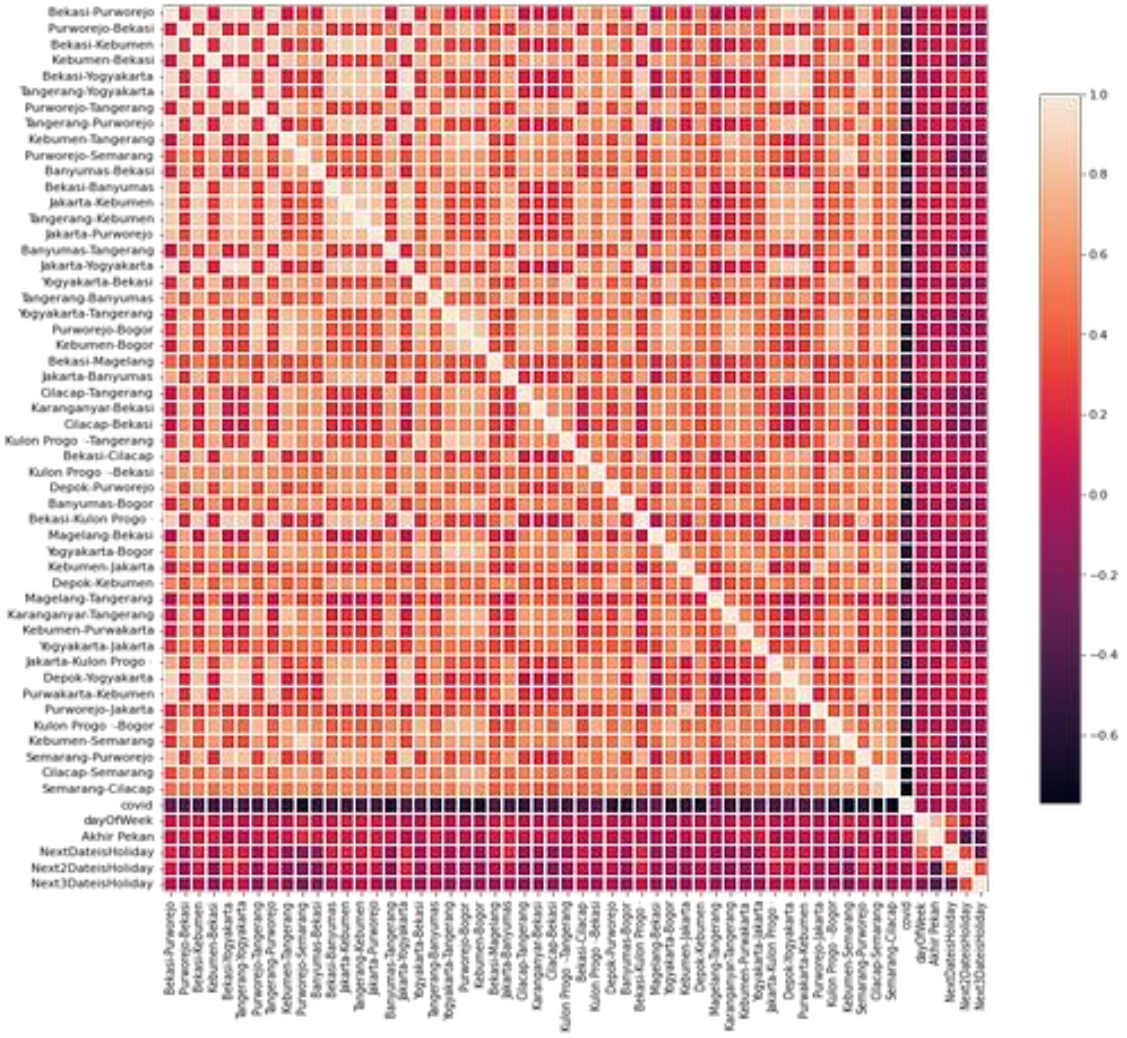

Figure 10. Heatmap

The value represents how each variable correlate with each other, 0 value means no correlation at all, 1 means the variable have strong positive correlation. This means when the value is going up the other value is going up as well with other variable, and negative value means the value has negative correlation with other value this means whenever variable value goes up, the value of other variables is going down since the correlation is negative.

Figure above shows that almost all variables have correlation with each other but not all variable correlate with each other with the same direction and this is normal, since each route will

have negative correlation with its return route for example yogyakarta-bekasi will have high negative correlation with bekasi-yogyakarata but yogyakarta bekasi probably only have weak correlation with semarang-kebumen either negative or positive.

Figure 11. Data distribution

From the data histogram from each route in Figure 11 shows that for all routes the data distribution is skew, this could make the deep learning model perform bad. From this, something must be done to make the data distribution have more Gaussian like shape (bell shape) like as mentioned in last chapter to achieve better deep learning models performance. To overcome this issue the author using power transformer from Scikit Learn library and pipeline from Scikit Learn, so with help of pipeline, the transformed data could be inverted back to its original form. In the

power transformer method, there are 2 transformer method which is box-cox method and Yeo-Johnson method and the method used is the Yeo-Johnson method since the box-cox method is strictly positive. That means box-cox method only allowing positive value. Although the data can used both methods, but in this study only Yeo-Johnson method will be used.

$$\psi(\lambda, y) = \begin{cases} ((y+1)^{\lambda} - 1)/\lambda & \text{if } \lambda \neq 0, y \geq 0 \\ \log(y+1) & \text{if } \lambda = 0, y \geq 0 \\ -[(-y+1)^{2-\lambda} - 1)]/(2-\lambda) & \text{if } \lambda \neq 2, y < 0 \\ -\log(-y+1) & \text{if } \lambda = 2, y < 0 \end{cases}$$

In this figure it can be seen that Yeo-Johnson formula have 4 conditional shapes, and with this the Yeo-Johnson have the ability to transform data that are not positive. For the result after using powertransformer with Yeo-Johnson formula, the data distribution is having more Gaussian like shape which is good news. The result can be seen in the figure below:

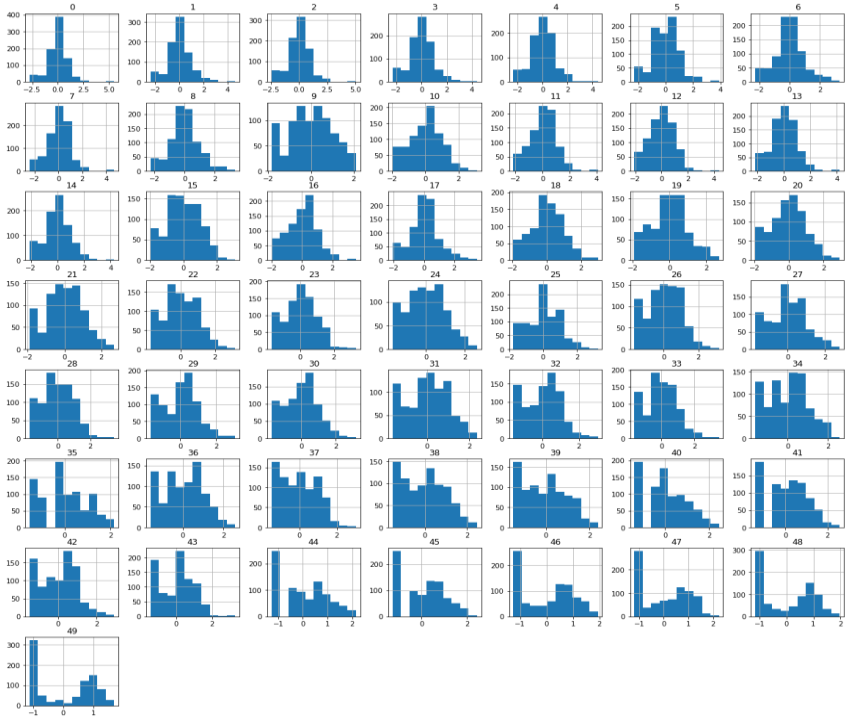

Figure 12. After using powertransformer

This method is for all the route variable but for the other variable ordinal scales and one hot encoder is used, ordinal encoder is mainly used to encode categorical data to numerical value, this is used for data that can be ordered, then each order could contribute for the training process such as dayofweek. When ordinal scales is use for dayofweek variable, it transforms the dayofweek value to numerical value (sunday to 0, monday to 1, etc). Then the last variable “namalibur” will

be transformed with one hot encoder. One hot encoder here will transform the data variable to binary values to get clear picture see figure below:

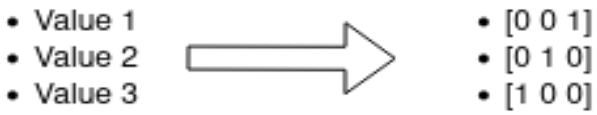

Figure 13. One Hot Encoder

The machine cannot understand words, so something must be done for the data, so the machine can understand it. One hot encoder and ordinal scaler is used to transform categorical data into numeric values, one hot encoder will transform data into binary models that can be seen in figure 4.9, then the ordinal scaler will transform data into ordered integer.

Deeplearning Model

In this research 2 Deep learning will be created then the performance will be compared from each other. The first Models are LSTM Autoencoder-Bi-LSTM Hybrid Models then the second one is Bi-LSTM models. In this study Keras model that will be used is the functional models, because this problem needs more than 1 output that cannot be handled by sequential API from Keras. This model then will be trained one by one first to save time then the trained models layer weight will be transferred for each combined model layer. This method is called transfer learning. The combined models will have multiple outputs.

LSTM Autoencoder-Bi-LSTM Hybrid

LSTM Autoencoders-Bi-LSTM model hybrid will have 2 part the first part is to create the LSTM autoencoders model and the second part will be creating Bi-LSTM for the output. Auto coders have 2 stages called the encoder stage then the decoder stages. Each stage will be stored into one function and act as layer. Encoder stages is a stage to compressing the data until it become into its latent space then from the latent space the decoder stages will be performed.

Figure 15 on the left is showing how the encoder models built, from those figures, its shows there will have 2 input, the first input is for the route data variable, and for the second one then both of them are concatenated together and will have data with 77 neurons. Those 77 neurons then go into encoding stages and leaving it into latent space with only 20 neurons, each neuron is representing the data dimensions. This is the end of the encoding stages; the output is data with its latent form with 20 variables. Then those 20 variables will then be inputted to decoding stages. The decoding stages will be combined with bi-LSTM in the end for the prediction output.

Figure 15 on the right shows the decoding-output models, start with the input which take from the encoder stages, then gradually rising the amount of variables then the output of decoder is directly inputted to bi-LSTM model with 20 neurons then the output will be inputted again into

another bi-LSTM then for the last the output from second bi-LSTM layer then will be inputted to dense layer which represent the prediction of single route variable.

Figure 16 is the architecture combined with 2 layers from 2 phases before which with functional model provided by Keras API, the models now have total of 5 output, each output represents single route predictions and each output have its own losses and metrics to get track of the model performance.

Bi-LSTM Model

This model is more simple than the LSTM-Autoencoders-Bi-LSTM hybrid model architecture. This model will directly input 2 data and output the prediction without any other architecture. This models will use bi-LSTM layers to create prediction.

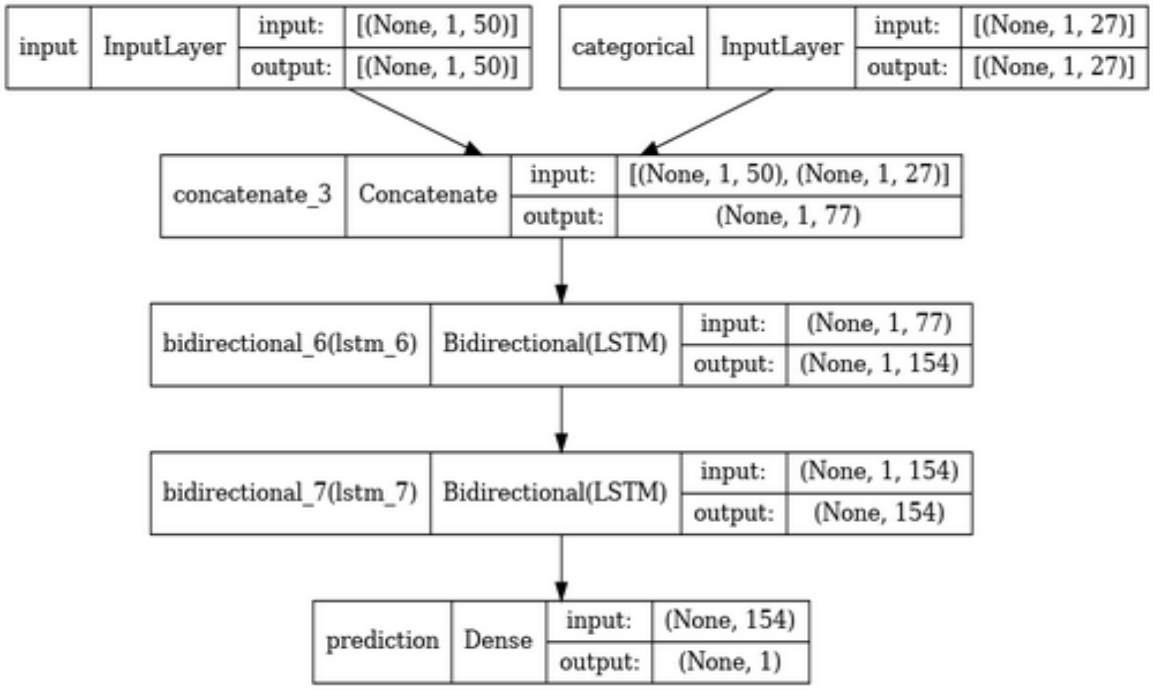

Figure 14. Bi-LSTM Model Layer

Like the LSTM-Autoencoders-Bi-LSTM hybrid, there will be 2 input node, each input represents the route variable and categorical data. Then those 2 input are concatenated into 1, then directly inputted to bi-LSTM layers with 77 neurons. Since the bidirectional learns data bidirectionally, so the number of neuron is also multiplicand by 2, That is why the output of 1 bidirectional layer is 154. The output from first bi-LSTM layer then inputted into second bi-LSTM layer which then the output is going into dense layer with 1 neuron that represents 1 single route.

This model will have 5 output as well that each output represents one route variable since in this research will only predict 5 route. Each output will have its own loss and metrics as well, so the performance can be tracked per output.

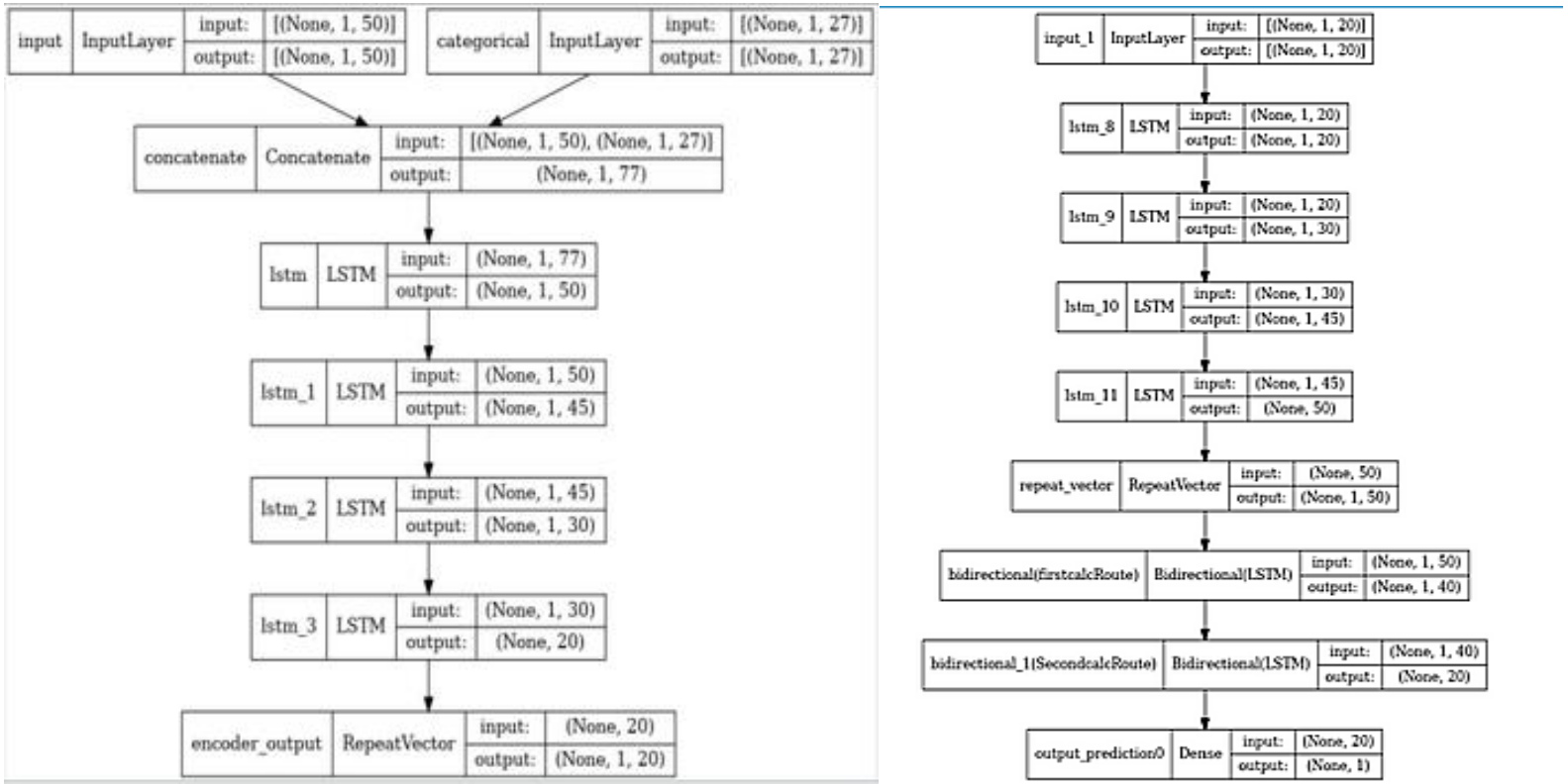

Figure 15. Left: LSTM encoder Layers, Right: decoder-output Layer

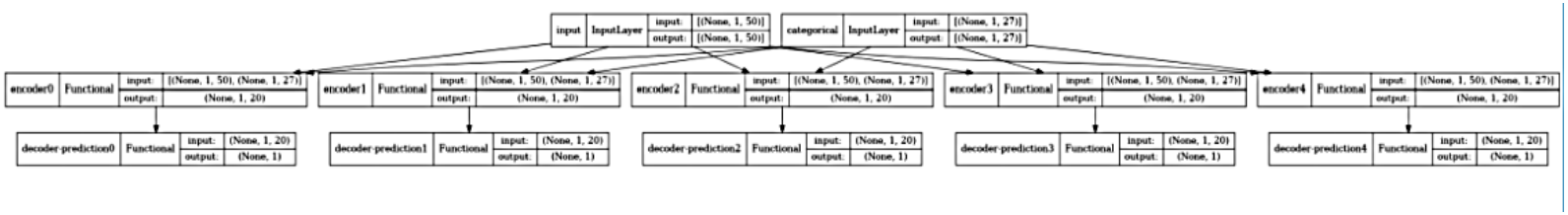

Figure 16. LSTM-Autoencoders-Bi-LSTM hybrid model architecture

RESULT

LSTM-Autoencoders-Bi-LSTM hybrid model

Training losses and metrics

ion used is MSE for the metrics RMSE is used. Lets se how LSTM-Autoencoders-Bi-LSTM hybrid model perform in figure below.

From figure above, we can see the loss and metrics value is gradually going down until the value is not going down anymore, the step of learning process is epoch, each epoch is trying to find the lowest loss and this is called global minima, to find the global minima, stochastic gradient descent is used for deep learning models with optimizer adam. Gradient desent is working by randomly set weight for each neuron to find the global minima as possible, so large epoch is needed. But, LSTM are facing problem of vanishing gradient and deep learning could overfit when deep learning models is trying to copy the input directly to outputs, and this is major problem, so to overcome this a limiter is need to get rid of vanishing gradient and overfitting. For this model the author use early stopping of the model training process with patience of 5, so when the epoch is not lowering anymore for 5 epochs, then the training process is automatically stops to prevent vanishing gradient and over fitting problem.

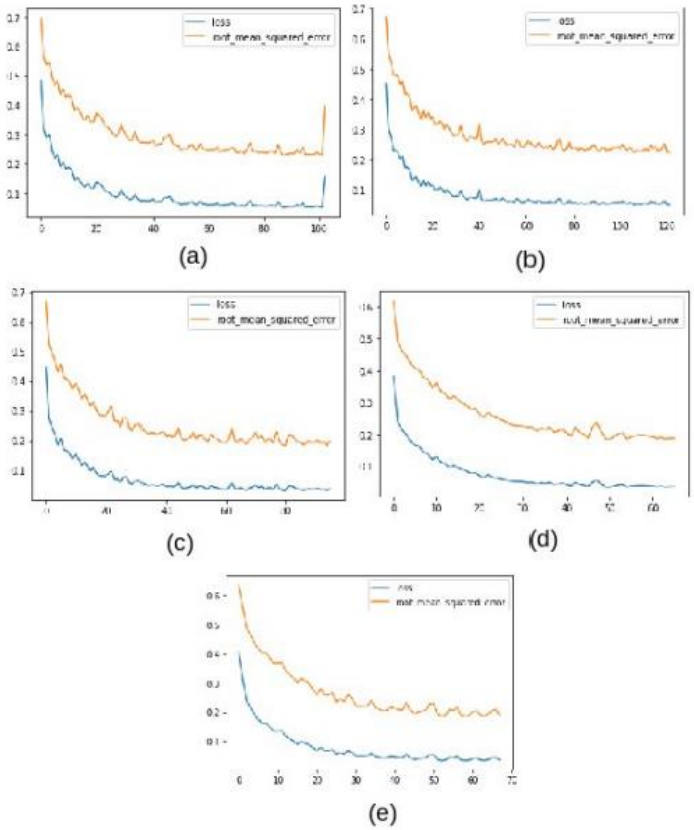

Figure 17. training Loss and metric (a) first route, (b) second route, (c) third route, (d) fourth route, (f) fifth route, for LSTM-Autoencoders-Bi-LSTM hybrid model

Table 1. Lowest Loss and metrics per route

	Route 1	Route 2	Route 3	Route 4	Route 5
Loss	0.0365	0.032	0.0288	0.0296	0.0273
Metrics	0.191	0.1789	0.1697	0.172	0.1652

As seen in table 1 all the loss value is below 0.1 and the metrics is below 1 Which is good news because the lower the loss and metrics value, the more similar the predicted value to the original value since MSE is the mean squared error of predicted by the original value.

Prediction of test set

In order to prove and judge about the model performance, testing of the trained model is needed So to prove test this models performance, the test set will be used to check the performance, the trained model then inputted test data and the models will try to predict the value per sequences but because the trained models input value is transformed with power transformer then to transform the data back just need to inverse the transformation back to its normal form. Then predicted result is plotted side by side with its training set value to proven that the model is performing well.

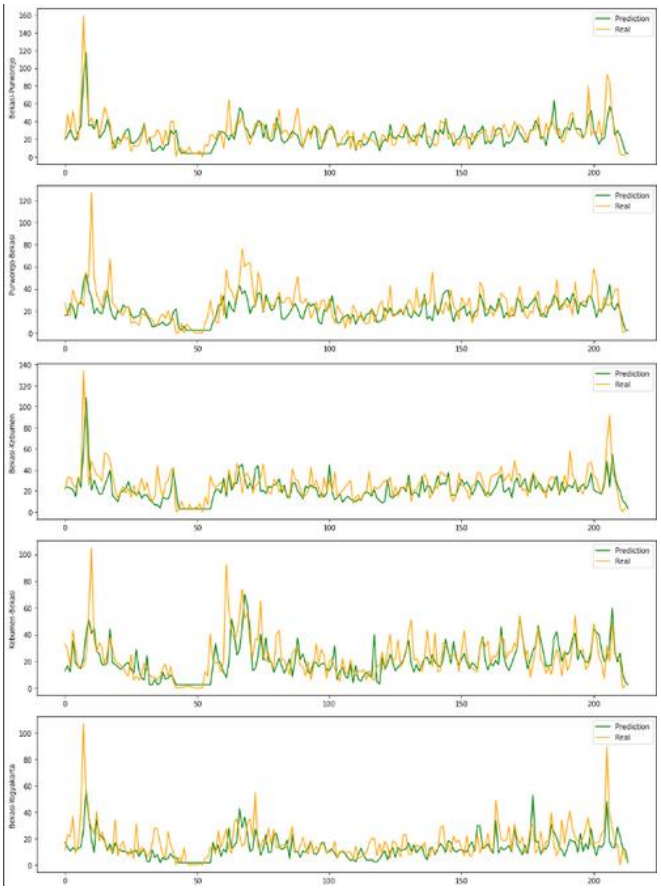

Figure 18. prediction result compared with real value

From Figure above, it shows that the prediction result (green line) when compared with the real value (yellow line), the result is very similar. This figure proves that the trained LSTM-Auto-encoders-Bi-LSTM hybrid model performs well enough and can predict the time-series data very well.

Bi-LSTM model

Training losses and metrics

To keep track of training process then loss function and metrics is used, the loss function used is MSE for the metrics RMSE is used. Let's see how Bi-LSTM model perform in figure below.

From figure above, we can see the loss and metrics value is gradually going down until the value is not going down any more just like the LSTM-Autoencoders-Bi-LSTM hybrid models, the step of learning process is epoch, each epoch is trying to find the lowest loss, and for this model the author use early stopping of the model training process with patience of 5 same as LSTM-Autoencoders-Bi-LSTM hybrid model, so when the epoch is not lowering anymore for 5 epochs, then the training process is automatically stop to prevent vanishing gradient and over fitting problem. As seen in table 2 most of the loss value is quiet similar to the models with autoencoders it means the Bi-LSTM models performance is as good as the models with autoencoders.

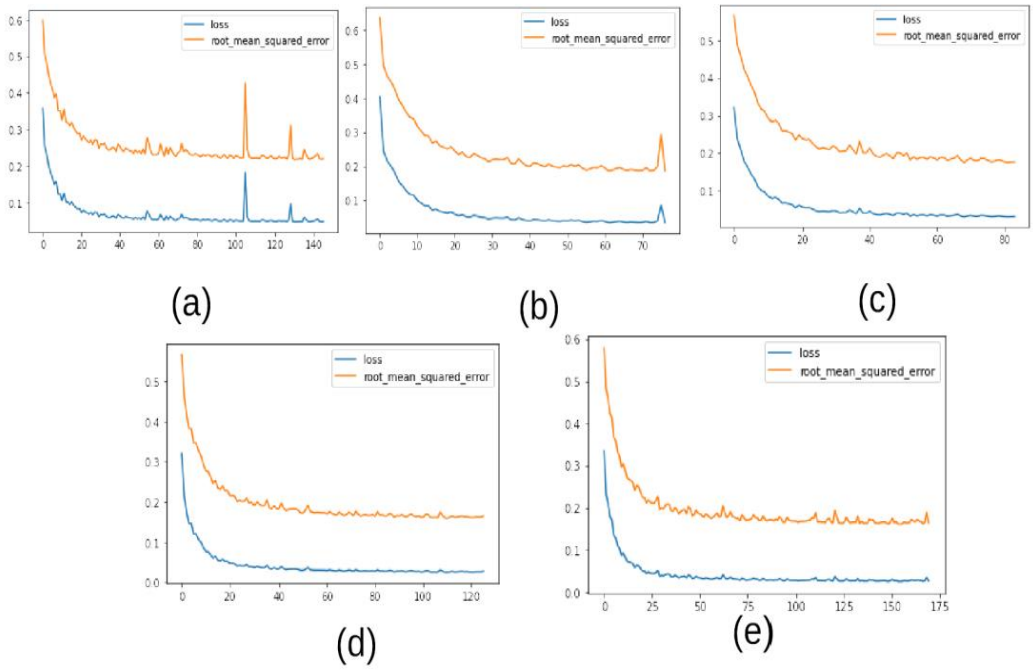

Figure 19. training Loss and metric (a) first route, (b) second route, (c) third route, (d) fourth route, (f) fifth route, for Bi-LSTM model

Table 2. Lowest Loss and metrics per route

	Route 1	Route 2	Route 3	Route 4	Route 5
Loss	0.0291	0.0354	0.027	0.0269	0.0208
Metrics	0.1706	0.1881	0.1643	0.164	0.1442

Prediction of test set

In order to prove and judge about the Bi-LSTM model performance, testing of the trained model is needed. So to prove test this models performance, the test set will be used to check the performance, the trained model then inputted test data and the models will try to predict the value per sequences just like what done for the models with autoencoder.

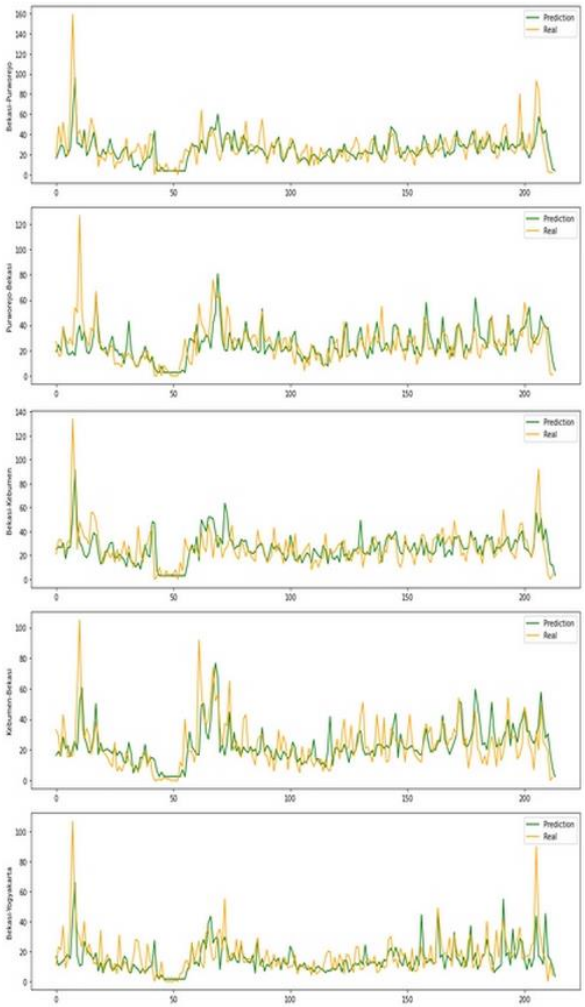

Figure 20. Prediction result compared with real value

From Figure above, it shows that the prediction result (green line) when compared with the real value (yellow line), the result is very similar. And compared to the models with auto encoder, the model's performance is also performs well for this problem.

Comparisons

From table 3 we can see that 2 models doesn't differ very much, from both of the models, the models without the autoencoder have the lowest MSE and RMSE value in route 5. Both models performs well, just a little difference between those 2 models.

Table 3. Model 1(**LSTM-Autoencoders-Bi-LSTM hybrid**), Model 2(Bi-LSTM)

	Route 1		**Route 2**		**Route 3**		**Route 4**		**Route 5**	
	Model 1	***Model 2***	***Model 1***	***Model 2***	***Model 1***	***Model 2***	***Model 1***	***Model 2***	***Model 1***	***Model 2***
Loss	0.0365	0.0291	0.032	0.0354	0.0288	0.027	0.0296	0.0269	0.0273	0.0208
Metrics	0.191	0.1706	0.1789	0.1881	0.1697	0.1643	0.172	0.164	0.1662	0.1442

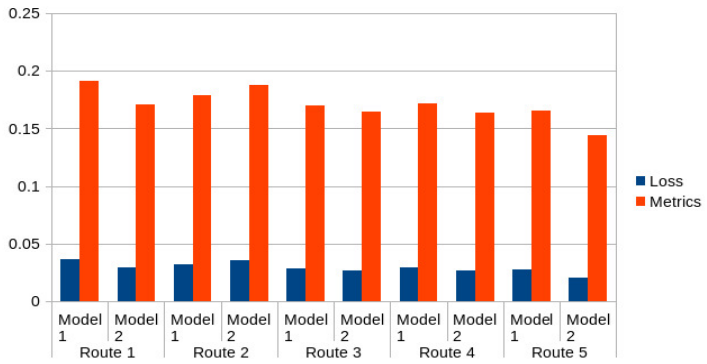

Figure 21. loss and metrics visualization for both models

From figure 22 we can see that model 1 (LSTM-Autoencoders-Bi-LSTM hybrid) have the largest loss with value of 0.0365, and Metrics with value 0.191 in route1 where for model 2 have largest value in route 2 with loss value 0.0354 and metrics 0.1881 which can said the largest loss and metrics between 2 model have difference 0.0011 for the loss and 0.0029 for the metrics. Model 2 (Bi-LSTM) have the lowest loss and metrics value with value of loss is 0.0208 and for the metrics is 0.1442 on route 5, when the lowest value for model 1 is also in route 5 with loss value of 0.0273 and metrics 0.1652 which can be said that the difference is 0.0065 for the loss and 0.021 for the metrics.

Table 4. Average Error

	Model 1	**Model 2**
Average MSE	0.03084	0.02784
Average RMSE	0.17536	0.16624

From figure above it shows the average MSE (Loss), and average RMSE (Metrics) from Model 1 and Model 2, from this figuse it now can clearly have better results than the model 1. That said that even with little difference the model 2 have better performance than the model 1 and 1 things that differ these 2 models the most is the training time, the autoencoders training time is

very slow compared to models without autoencoders. This is normal for autoencoder to train slower than without it due to more network depth of the models with autoencoder.

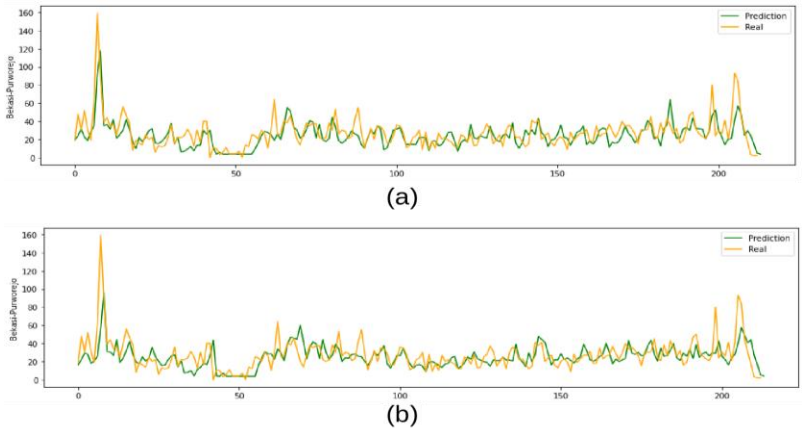

Figure 22. prediction result with autoencoder (a), without autoencoder (b)

As shown in figure above, although the model 2 is overcome the model 2, from figure above we can see there are no massive differences with the models with autoencoder nor without auto encoders. Both of the models perform really well for these problems. Probably when the data variable is larger than from this study, the auto encoder could perform much better than from this study.

CONCLUSION

From this Bus Route Demand Prediction With Deep Learning study LSTM-Autoencoders-Bi-LSTM hybrid models and Bi-LSTM is created to get comparison from those 2 models, then ftom this research it can be concluded that :

1. with huge amount of data that the authors get, surge in demand for bus routes is predictable since all deep learning model created in this study could predict the bus route demand decently thanks to the LSTM layers, this time series problem could solved.
2. Many other data can be extracted from the raw data that can help deep leraning training stages, from this reseach the author could extract 'covid', 'dayOfWeek', 'Akhir Pekan', "NextDateisHoliday", "Next2DateisHoliday", "Next3DateisHoliday", out from the date time variable only, so the trainable deep learning models may have good results.
3. Autoencoders in their nature will do every job very well on extraxting features out from high dimensionality of the data, since the autoencoders it self is used to use on unsupervised learning problems. Autoencoders also done a good job for extracting feature from this study. The encoding step is missing some information, but the information that loss is information that doesent realy important for the prediction so the machine can get more information that more meaning full and drop the unwanted one automaticly. But in this study it shows that models with autoencoders when compared with models that dosent have autoencoders dosent differ that much. This happen probably because the data dimension in

this study is not large enough so that without autoencoder the models can predict pretty well. Its also need to be noted that autoencoder have their own drawbacks due to its deeper network that hugely impact the training time.

And here is for the suggestion and adfice from the author for future research:

1. Due to the lack of resouce and time that the author have, then further research for this study is needed to prove more then from this research, probably adding more data variable to se how it will impact for the models.
2. Adding more deep learning models is needed so more than 1 models could compared each from each other, probably using 2d Convolutional neural networks, LSTM-CNN hybrid or even using machine learning method such as SVM.
3. Using more powerfull machine to hugely cut of the training time so more models could created for comparisons.

DAFTAR PUSTAKA

[1] S. A. Krishnaswamy, M. Paul, and Srivatsa Krishnaswamy, "A Comparative Study and Analysis of Time Series Forecasting Techniques | Enhanced Reader," 2020. Available: https://www.researchgate.net/publication/337244936_Comparative_Study_on_Time_Series_Forecasting_Models.

[2] K. Yang and C. Shahabi, "A pca-based kernel for kernel pca on multivariate time series," Proc. ICDM 2005 Work. …, no. June, 2005, [Online]. Available: http://w2.math.bme.hu/kanya/astor/tdm05.pdf.

[3] K. Nazmoon, T. Tahmid, A. Rafi, and M. Ehsanul, "Forecasting COVID-19 cases: A comparative analysis between recurrent and convolutional neural networks," no. January, 2020. Available: https://www.researchgate.net/publication/350978557_Forecasting_COVID-19_cases_A_comparative_analysis_between_recurrent_and_convolutional_neural_networks.

[4] J. G. Taylor, "Univariate and Multivariate Time Series Predictions," no. January, pp. 11–22, 2002, doi: 10.1007/978-1-4471-0151-2_2. Available: https://www.sciencegate.app/app/document/download/10.1007/978-1-4471-0151-2_2

[5] I. Sülo, Ş. R. Keskin, G. Doğan, and T. Brown, "Energy Efficient Smart Buildings: LSTM Neural Networks for Time Series Prediction," Proc. - 2019 Int. Conf. Deep Learn. Mach. Learn. Emerg. Appl. Deep. 2019, pp. 18–22, 2019, doi: 10.1109/Deep-ML.2019.00012. Available: https://ieeexplore.ieee.org/document/8876919.

[6] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, "Recurrent Neural Networks for Multivariate Time Series with Missing Values," Sci. Rep., vol. 8, no. 1, Dec. 2018, doi: 10.1038/s41598-018-24271-9.

[7] K. Ouyang, Y. Hou, S. Zhou, and Y. Zhang, "Convolutional neural network with an elastic matching mechanism for time series classification," Algorithms, vol. 14, no. 7, 2021, doi: 10.3390/a14070192. Available: https://mdpi-res.com/d_attachment/algorithms/algorithms-14-00192/article_deploy/algorithms-14-00192.pdf

[8] Y. Wen, P. Lin, and X. Nie, "Research of stock price prediction based on PCA-LSTM model," IOP Conf. Ser. Mater. Sci. Eng., vol. 790, no. 1, 2020, doi: 10.1088/1757-

899X/790/1/012109. Available: https://www.researchgate.net/publication/340490809_Research_of_Stock_Price_Prediction_Based_on_PCA-LSTM_Model
[9] Y. Yuan et al., "Bus dynamic travel time prediction: Using a deep feature extraction framework based on rnn and dnn," Electron., vol. 9, no. 11, pp. 1–20, 2020, doi: 10.3390/electronics9111876. Available: https://www.mdpi.com/2079-9292/9/11/1876
[10] L. Badal and S. Franzén, "A Comparative Analysis of RNN and SVM Electricity Price Forecasting in Energy Management Systems," DEGREE Proj. Comput. Eng., 2019. Available: http://kth.diva-portal.org/smash/record.jsf?pid=diva2%3A1353342&dswid=-602