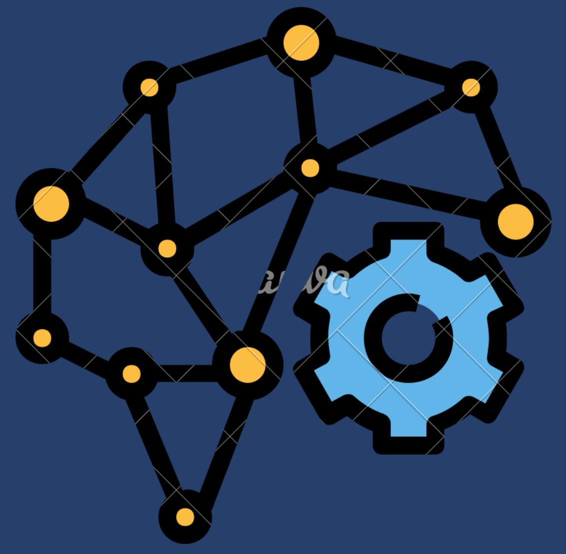
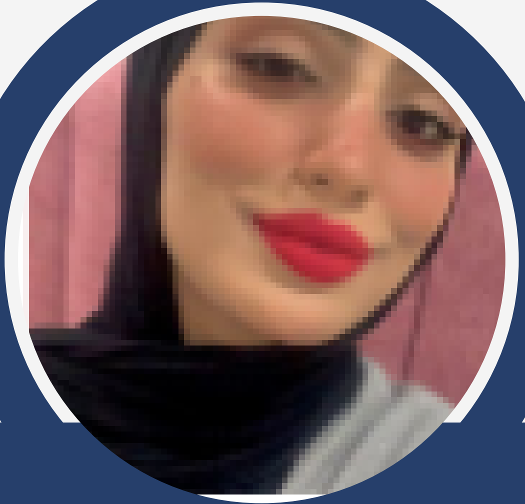
DEEP-LEARNING

NEURAL NEXUS

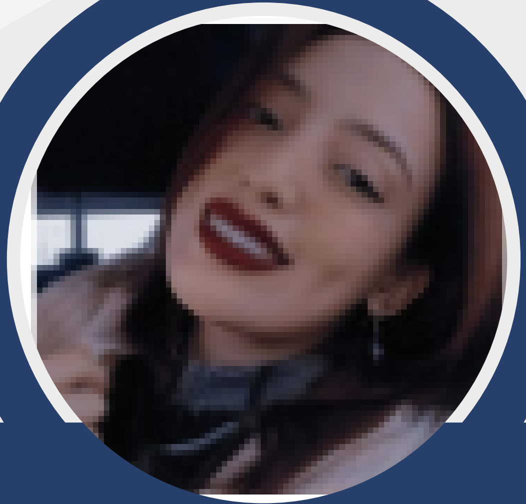


12/05/2023

TEAM



Marwa HAJ AYED



Eya BEN JEMAA

Armand Bryan FOZAMEENDEZOUMOU

Abdessalem DRINE

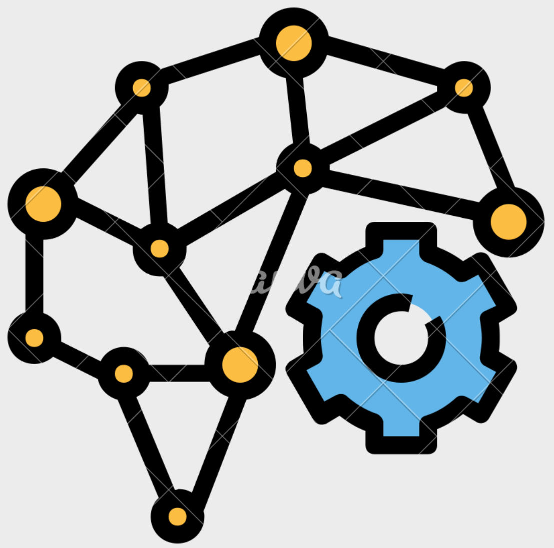
TABLE OF CONTENTS

INTRODUCTION

Traffic Forecasting in 5G Network using LSTM

PROBLEM

Accurately predicting 5G traffic to optimize resource allocation and enhance network performance.



STUDY OF THE EXISTING

Traffic Prediction in 5G Networks Using LSTM Neural Networks

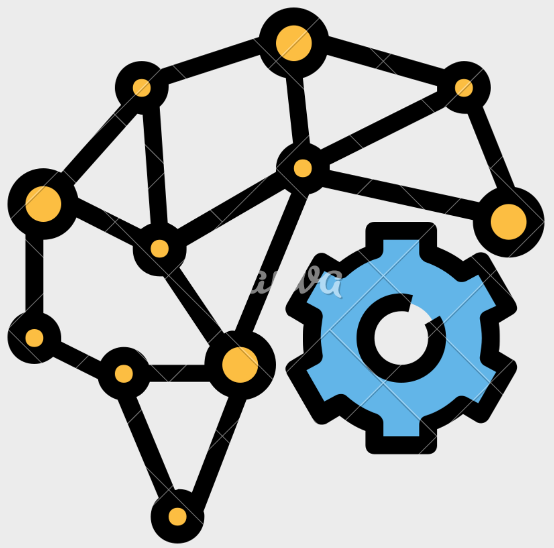
THIS STUDY EXPLORES THE USE OF LSTM NEURAL NETWORKS FOR PRECISE TRAFFIC PREDICTION IN 5G NETWORKS. BY CAPTURING COMPLEX TRAFFIC PATTERNS, LSTM MODELS OPTIMIZE RESOURCE ALLOCATION EFFECTIVELY. WITH REAL TRAFFIC DATA, THE STUDY DEMONSTRATES THE POTENTIAL OF LSTM IN ENHANCING TRAFFIC PREDICTION ACCURACY AND IMPROVING OVERALL NETWORK PERFORMANCE IN 5G ENVIRONMENTS.

A Comparative Study of Traffic Prediction Models in 5G Networks

THIS COMPARATIVE STUDY FINDS THAT LSTM, AN RNN-BASED MODEL, OUTPERFORMS OTHER METHODS IN ACCURATELY PREDICTING TRAFFIC IN 5G NETWORKS. THESE RESULTS HIGHLIGHT THE IMPORTANCE OF ADVANCED MODELS FOR OPTIMIZING RESOURCE MANAGEMENT AND IMPROVING TRAFFIC PREDICTION IN 5G NETWORKS.

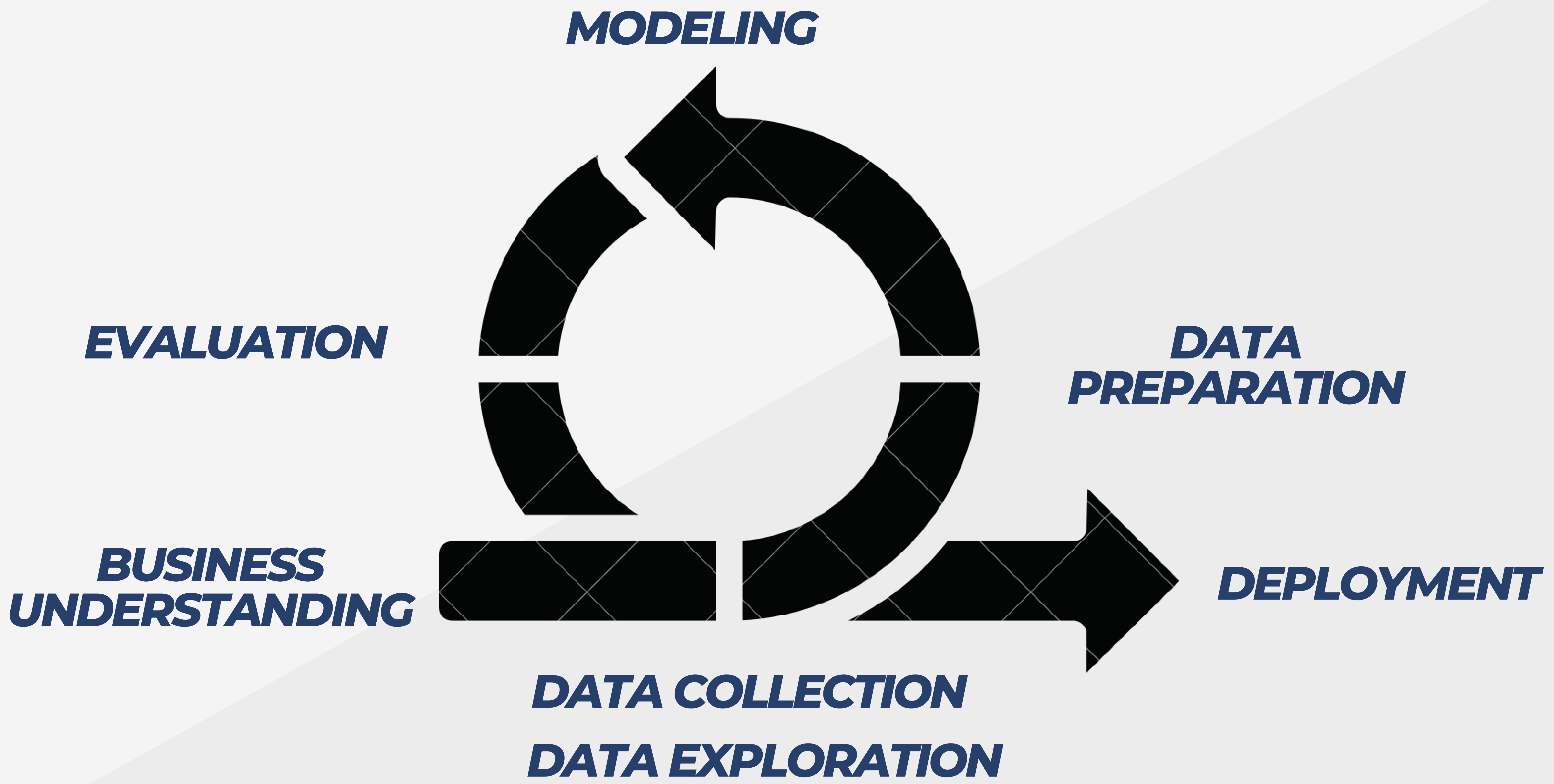
SOLUTION

Using deep learning algorithms for traffic prediction in 5G networks.



CRISP-DM METHOD

CRISP-DM METHODOLOGY



DATA EXPLORATION

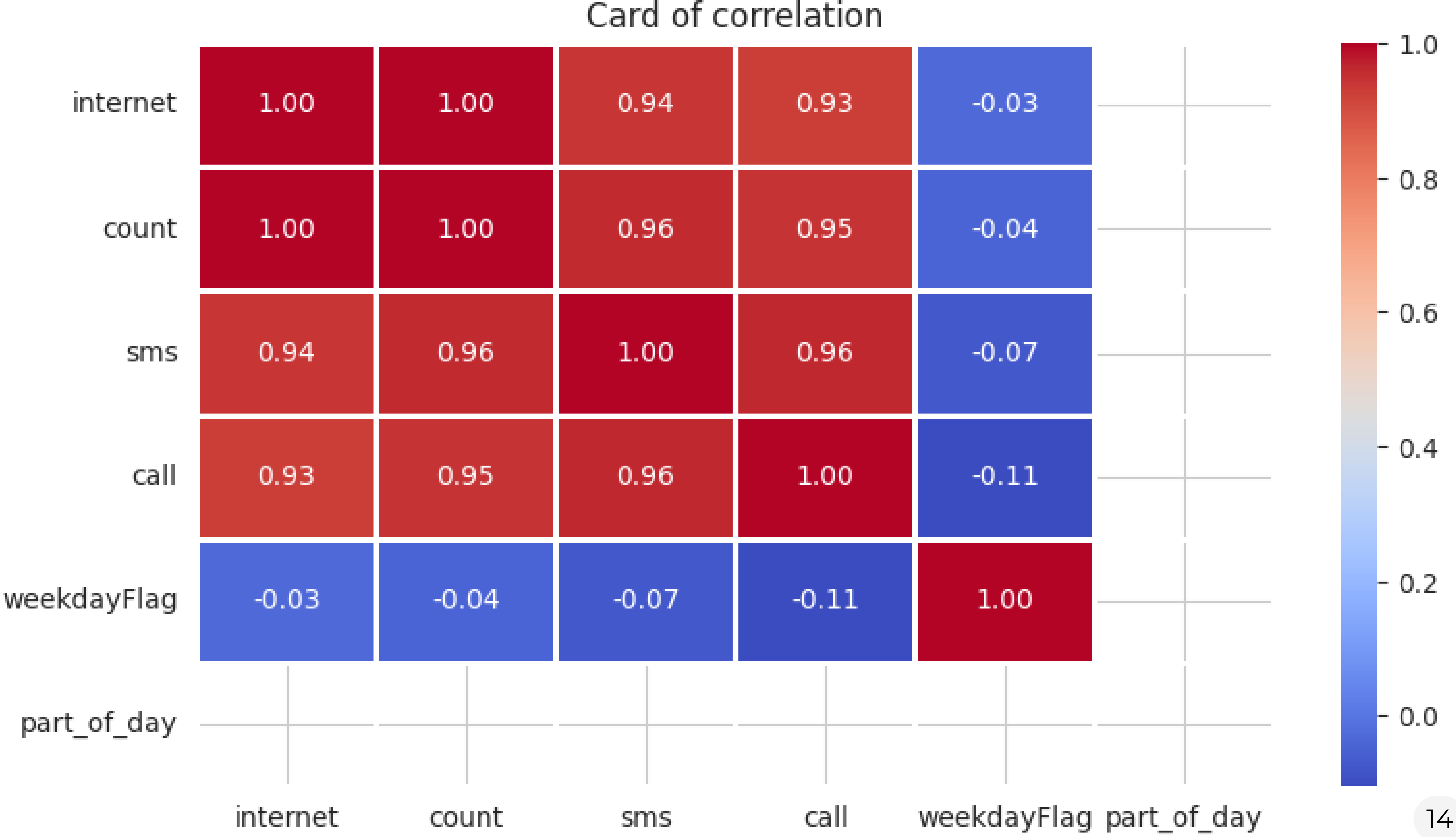
	gridID	startTime	smsIn	smsOut	callIn	callOut	internet
0	1	2013-11-01	78.709755	45.886570	41.108567	48.245378	1507.048349
1	1	2013-11-02	86.415810	43.875946	47.891016	53.590637	1515.641856
2	1	2013-11-03	77.728292	45.446780	36.145436	40.906425	1533.148425
3	1	2013-11-04	104.793806	54.821018	67.898464	70.399418	1404.813593
4	1	2013-11-05	97.425105	46.607029	68.735213	70.766221	1518.090111
...	...	...	...	...	...	...	...
619928	10000	2013-12-28	177.422546	71.416895	108.930903	131.882476	2373.501572
619929	10000	2013-12-29	130.030895	56.847904	83.989946	98.134224	2452.266242
619930	10000	2013-12-30	178.314106	71.122147	143.387445	158.163057	2198.423026
619931	10000	2013-12-31	259.851232	141.173667	137.291843	159.614859	2256.632144
619932	10000	2014-01-01	254.696088	152.534149	106.231598	126.834037	2196.453302

619933 rows × 7 columns

DATA PREPARATION

```python
#print sample of new data
dailyGridActivity.sample(4)
```

		internet	count	sms	call	weekdayFlag	holiday	part_of_day
gridID	**startTime**							
2084	**2013-12-31**	2533.845203	3147.465982	326.146930	287.473849	1	0	0
5993	**2013-11-02**	1455.324340	1826.317115	206.707502	164.285273	5	1	0
3909	**2013-11-16**	1663.609311	1880.796160	105.002754	112.184094	5	0	0
7770	**2013-12-09**	25880.407751	30766.389733	2665.301925	2220.680056	0	0	0

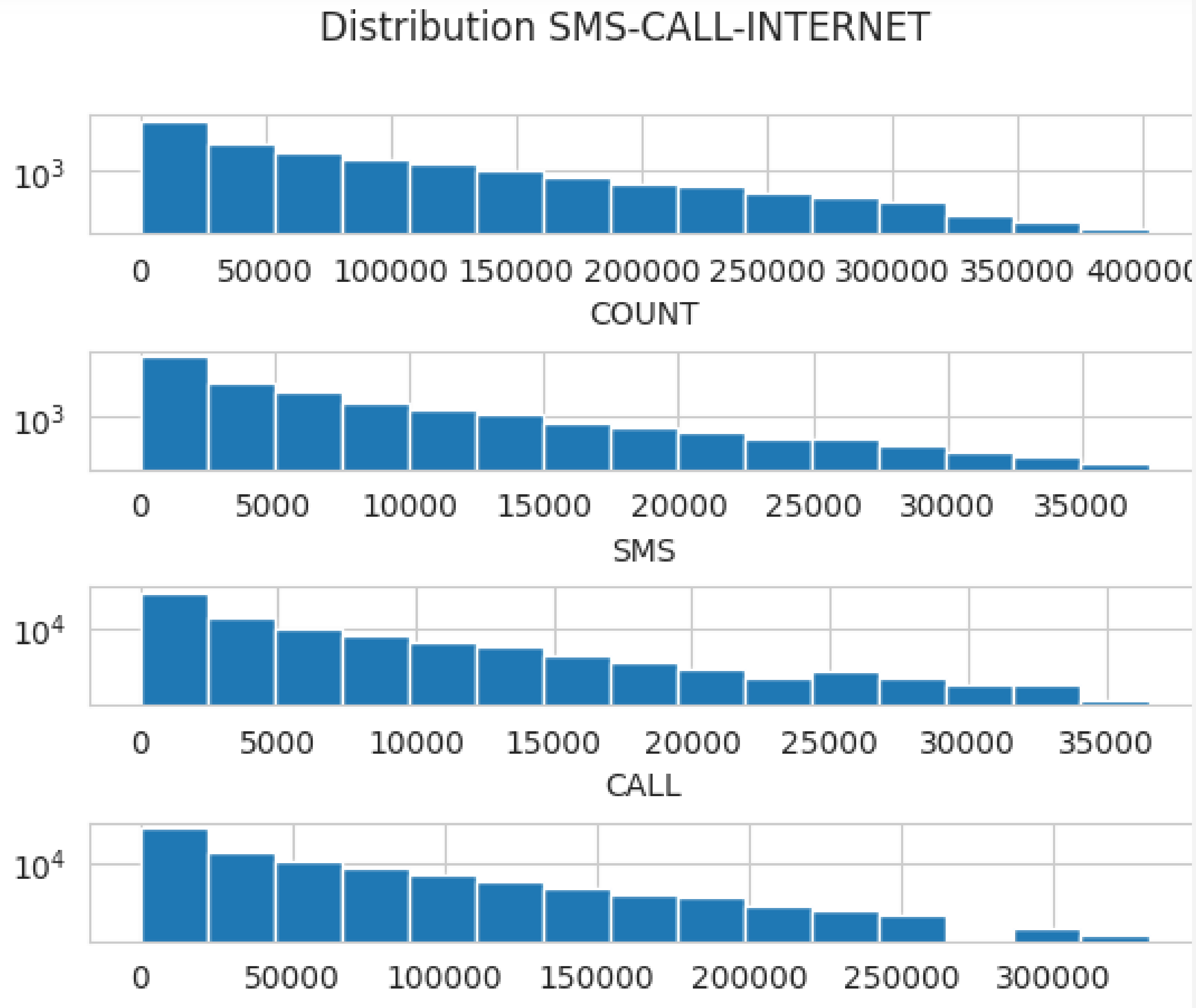

Card of correlation
internet
count
sms
call
weekdayFlag
part_of_day
1.00
1.00
0.94
0.93
-0.03
1.00
1.00
0.96
0.95
-0.04
0.94
0.96
1.00
0.96
-0.07
0.93
0.95
0.96
1.00
-0.11
-0.03
-0.04
-0.07
-0.11
1.00
internet
count
sms
call
weekdayFlag
part_of_day
1.0
0.8
0.6
0.4
0.2
0.0

DESCRIBE METHOD

```
dailyGridActivity.describe().T
```

	count	mean	std	min	25%	50%	75%	max
internet	619933.0	8957.248908	15376.536548	0.000000	1782.880566	4254.068418	9188.972982	331244.215800
count	619933.0	10894.746262	18578.251208	0.300023	2197.856144	5254.453316	11159.280239	401991.571896
sms	619933.0	1008.488025	1777.567264	0.000000	183.464545	466.154523	1032.067081	37442.964641
call	619933.0	929.009329	1610.637689	0.000000	170.363478	436.875254	978.965350	36542.184142
weekdayFlag	619933.0	3.000002	2.016063	0.000000	1.000000	3.000000	5.000000	6.000000
part_of_day	619933.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

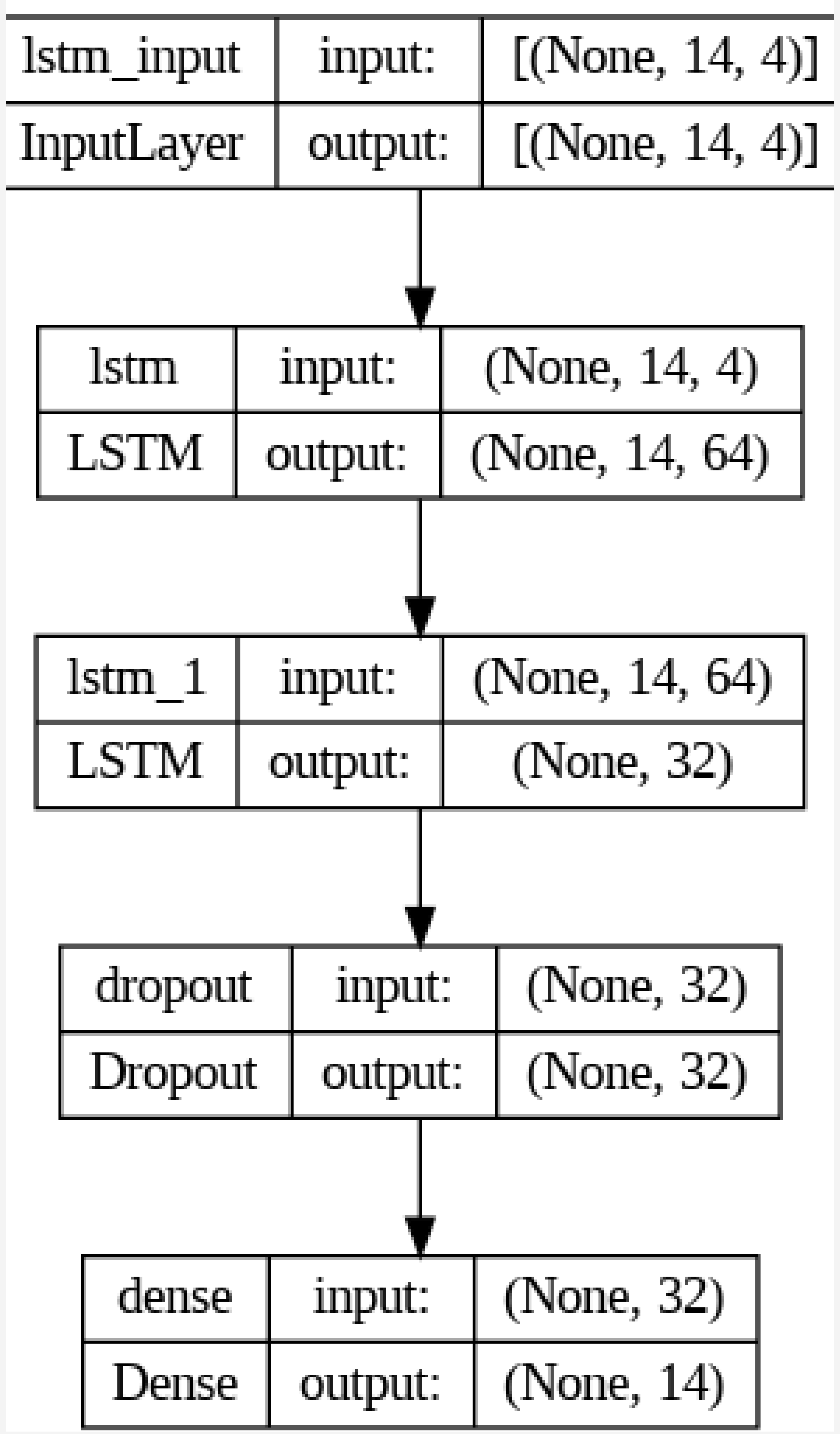
DISTRIBUTION OF THE DATA



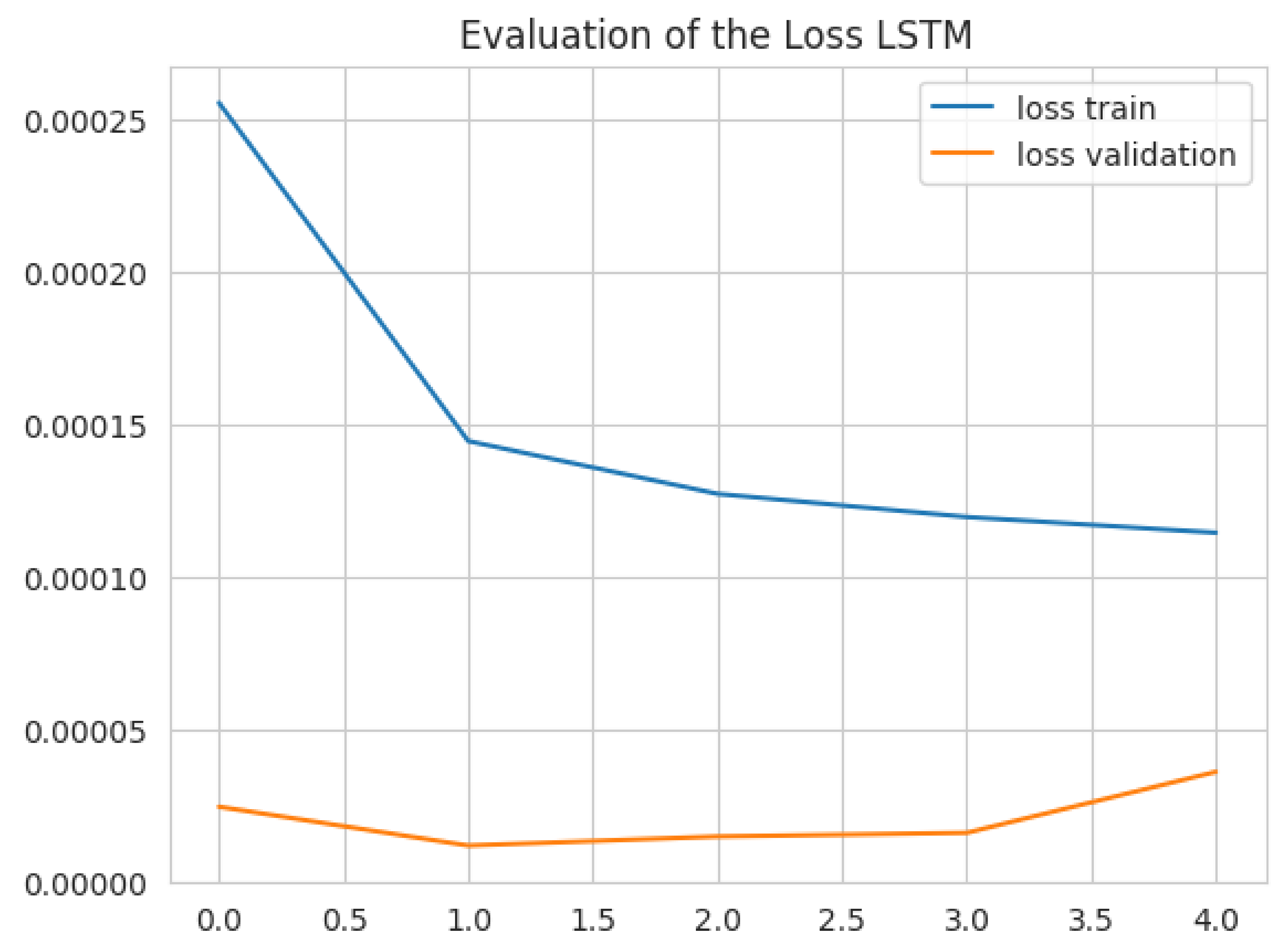
MODELING

- LSTM
- GRU
- CONV1D

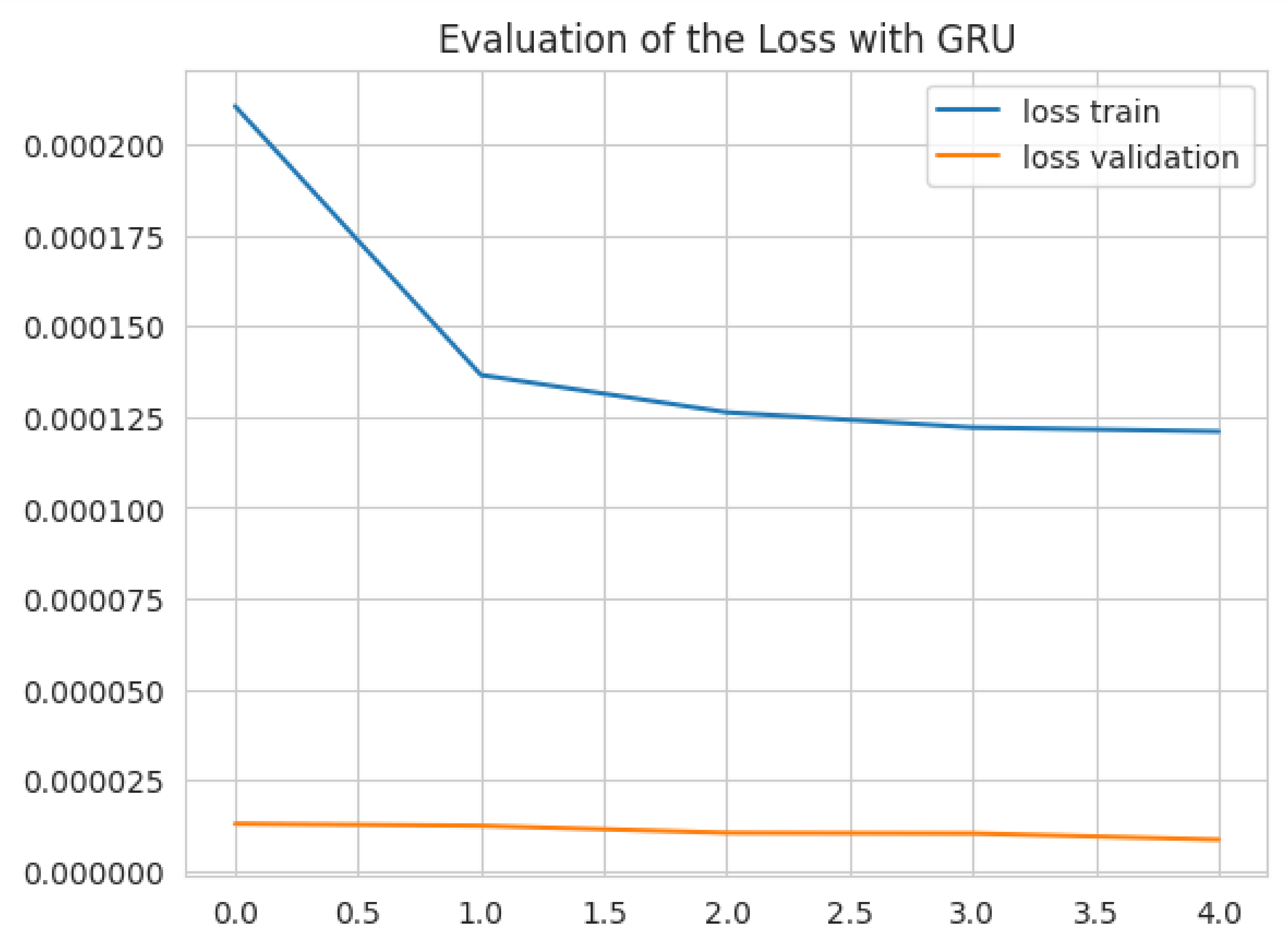
LSTM ARCHITECTURE



LSTM RESULT


Evaluation of the Loss LSTM
loss train
loss validation
0.00025
0.00020
0.00015
0.00010
0.00005
0.00000
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

GRU RESULT


Evaluation of the Loss with GRU
loss train
loss validation
0.000200
0.000175
0.000150
0.000125
0.000100
0.000075
0.000050
0.000025
0.000000
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

CONV1D LSTM ARCHITECTURE

```
[ ]  model2.summary()

Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d_1 (Conv1D)           (None, 9, 256)            6400

 max_pooling1d_1 (MaxPooling  (None, 4, 256)           0
 1D)

 lstm_2 (LSTM)               (None, 128)               197120

 flatten_1 (Flatten)         (None, 128)               0

 dense_4 (Dense)             (None, 192)               24768

 dense_5 (Dense)             (None, 1)                 193

=================================================================
Total params: 228,481
Trainable params: 228,481
Non-trainable params: 0
```

CONV1D LSTM RESULT

```
history_cnn1d = model2.fit(trainx_tensor,trainy_tensor, epochs=5, batch_size=100, validation_split=0.1, verbose=1, callbacks=[tensorboard_callback])
```

```
Epoch 1/5
5580/5580 [==============================] - 180s 32ms/step - loss: 1.5546e-04 - mean_absolute_error: 0.0052 - val_loss: 1.4374e-05 - val_mean_absolute_erro
Epoch 2/5
5580/5580 [==============================] - 205s 37ms/step - loss: 1.1579e-04 - mean_absolute_error: 0.0043 - val_loss: 2.8208e-05 - val_mean_absolute_erro
Epoch 3/5
5580/5580 [==============================] - 180s 32ms/step - loss: 1.0516e-04 - mean_absolute_error: 0.0040 - val_loss: 1.2784e-05 - val_mean_absolute_erro
Epoch 4/5
5580/5580 [==============================] - 168s 30ms/step - loss: 9.9746e-05 - mean_absolute_error: 0.0039 - val_loss: 1.3179e-05 - val_mean_absolute_erro
Epoch 5/5
5580/5580 [==============================] - 240s 43ms/step - loss: 9.5659e-05 - mean_absolute_error: 0.0038 - val_loss: 1.1234e-05 - val_mean_absolute_erro
```

```
loss_cnnlstm = np.mean(history_cnn1d.history['mean_absolute_error'])
val_loss_cnnlstm = np.mean(history_cnn1d.history['val_mean_absolute_error'])

print(' The Loss of the Conv1d LSTM model training = {} \n The Loss on validation data with Conv1d LSTM model = {}'.format(loss_cnnlstm, val_loss_cnnlstm))
```

```
 The Loss of the Conv1d LSTM model training = 0.004240116290748195
 The Loss on validation data with Conv1d LSTM model = 0.0024808562127873303
```


EVALUATION

```
[ ] data_model = {'LSTM': loss_lstm, 'GRU':loss_gru, 'Conv1DLSTM':loss_cnnlstm}
```

```
[ ] pd.DataFrame(data=data_model, index=[0])
```

	LSTM	GRU	Conv1DLSTM
0	0.005341	0.005231	0.00424

Then see all val loss of the 3 models.

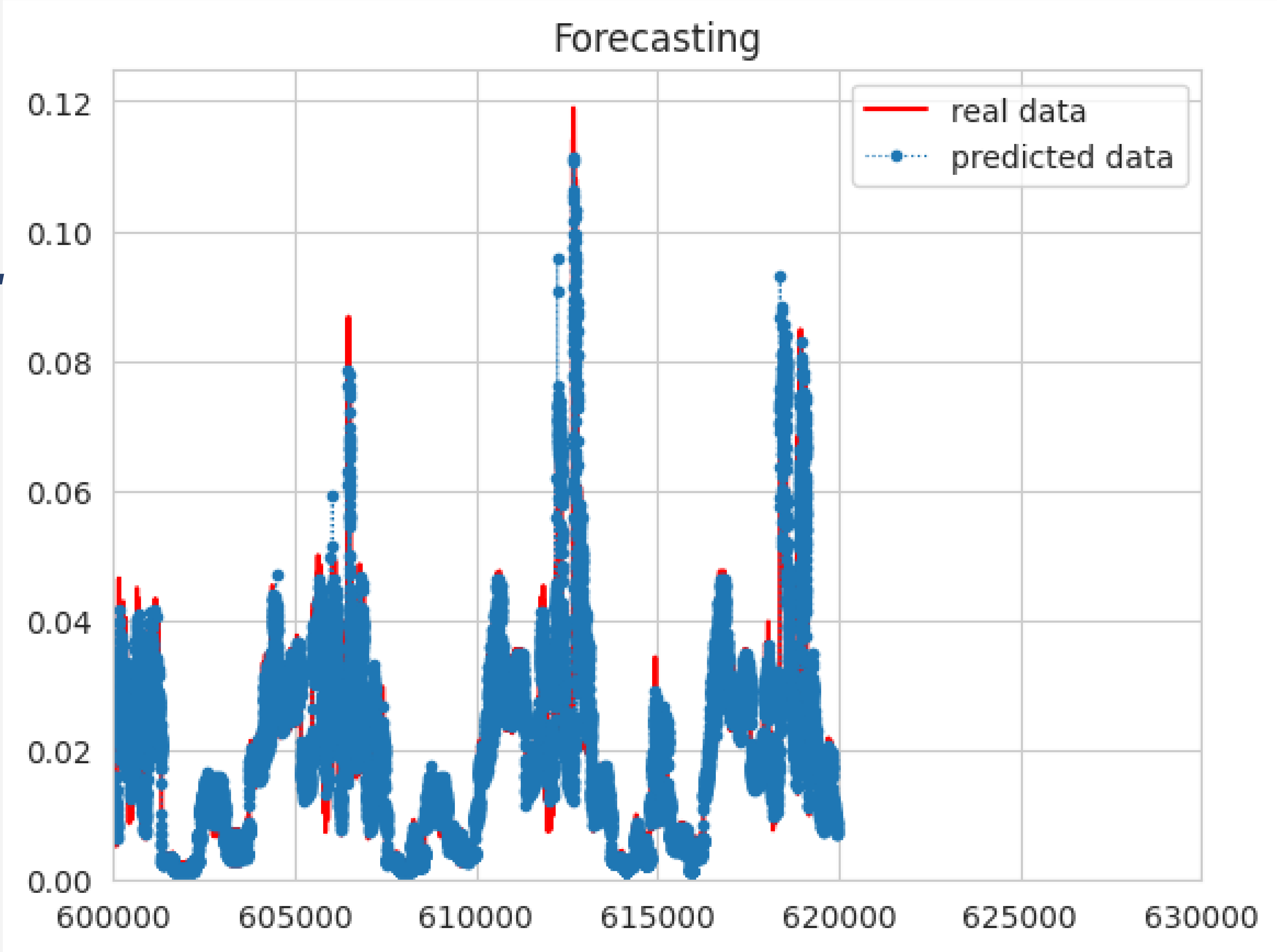
Add text cell

```
[ ] val_data_model = {'LSTM': val_loss_lstm, 'GRU':val_loss_gru, 'Conv1DLSTM':val_loss_cnnlstm
```

```
pd.DataFrame(data=val_data_model, index=[0])
```

	LSTM	GRU	Conv1DLSTM
0	0.003274	0.002113	0.002481

EVALUATION OF PREDICTED DATA AGAINST REAL DATA.



FINAL RESULT PREDICTED USING CONVID LSTM

```
Prediction of the next day (02-January-2014):

internet count sms call
array([[2702.53346836, 3280.03990818,  305.48719123,  298.13796269]])
```

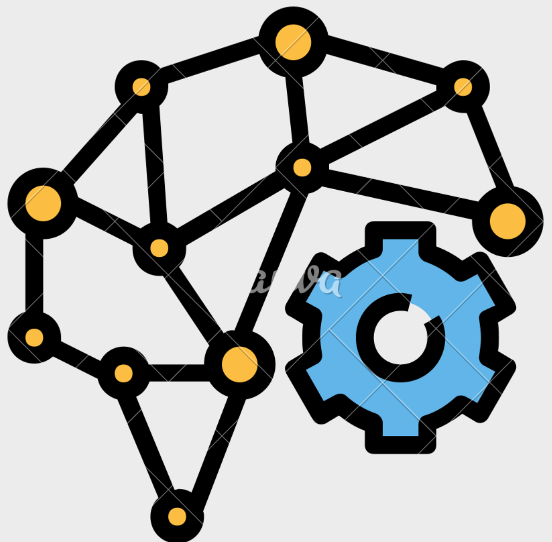
DEPLOYMENT

IMPROVEMENT OF OUR SOLUTION

Enhancing our deep learning-based traffic prediction system for 5G networks and potentially incorporating fault detection algorithms for proactive identification of network failures.

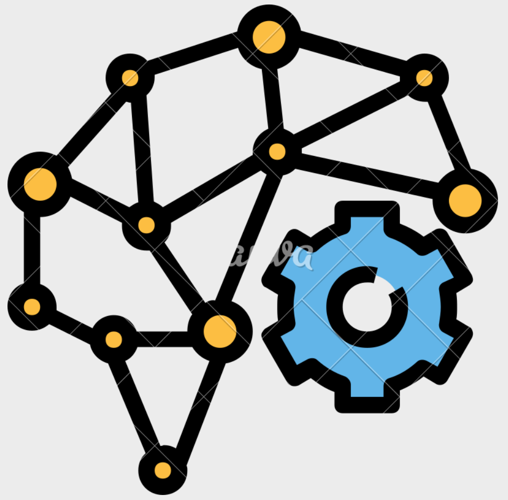
CONCLUSION

In summary, our project highlights the effectiveness of LSTM, GRU and Conv1D LSTM models in accurately predicting traffic in 5G networks. These advanced algorithms offer improved accuracy and resource optimization capabilities, making them valuable tools for network planning and optimization.



REFERENCE

- Show the market analysis of your company’s competitors
- https://ir.lib.uwo.ca/cgi/viewcontent.cgi?article=11545&context=etd
- https://www.hindawi.com/journals/cin/2022/3174530/
- https://www.researchgate.net/publication/369450347_A_comparative_study_of_cellular_traffic_prediction_mechanisms



THANK YOU