Deep Learning Workshop – Assignment 2 Report

Time Series

1. Exploratory Data Analysis:

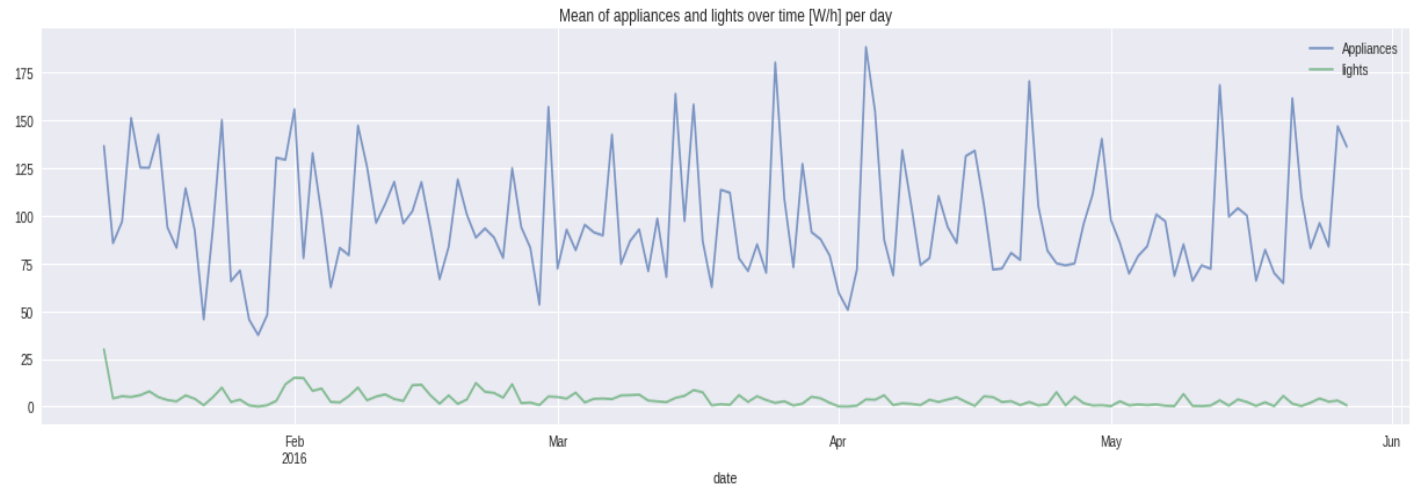
בחרנו את מאגר הנתונים של "Appliances Energy Prediction", שמכיל נתונים כמו טמפרטורות ואחוזי לחות בחדרים שונים בבית ומחוצה לו, ובנוסף גם את הכמות (בוואט) של האנרגיה שמשתמשים מכשירי החשמל והמנורות בבית. הדגימות נלקחו באינטרוול זמן של 10 דקות בין דגימה לדגימה.

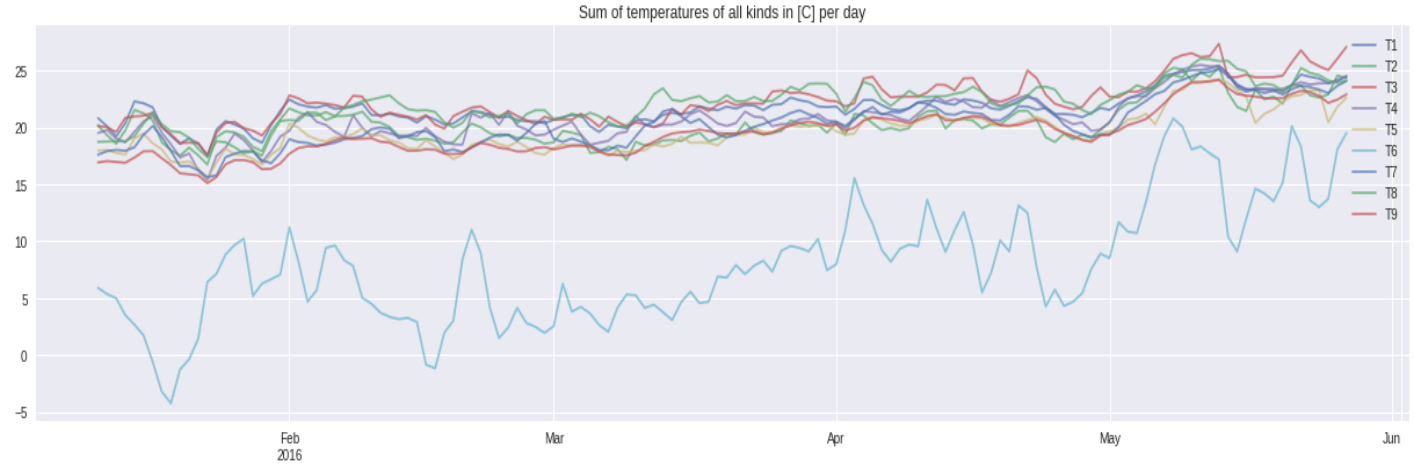
המאגר לא קובע מראש את מה אנחנו רוצים לחזות, לכן נקבע אנחנו שמשתנה המטרה שלנו יהיה Appliances, כלומר האנרגיה בשימוש ע"י מכשירי החשמל בבית (בוואט לשעה). מכן נובע שמדובר בבעיית רגרסיה – אנו מנסים לחזות את צריכת החשמל של המכשירים בבית (בוואט לשעה), שזה מספר טבעי (אמור להיות רציף אבל בנתונים של המאגר הוא טבעי שלם).

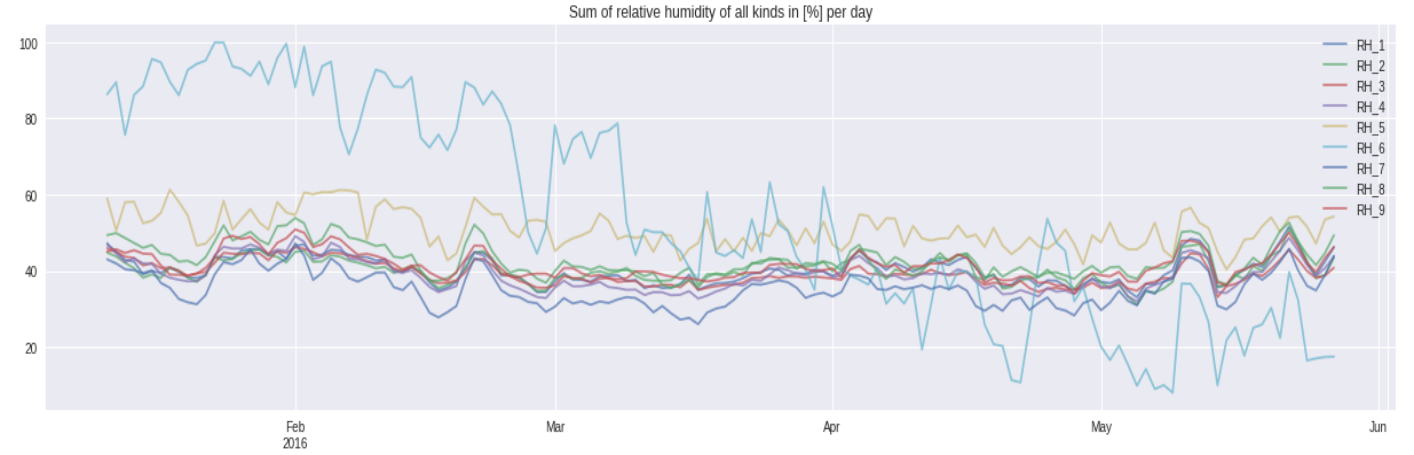
מאגר הנתונים שלנו מורכב מ-19735 דוגמאות ו-29 פיצ'רים. נסתכל על דוגמאות מהמאגר:

```
#### Our data has 19735 number of examples and 28 features.
#### Number of missing values in our entire data: 0
#### A quick look at our data:


                     Appliances  lights     T1       RH_1    T2       RH_2  \
date
2016-01-11 17:00:00          60      30  19.89  47.596667  19.2  44.790000
2016-01-11 17:10:00          60      30  19.89  46.693333  19.2  44.722500
2016-01-11 17:20:00          50      30  19.89  46.300000  19.2  44.626667
2016-01-11 17:30:00          50      40  19.89  46.066667  19.2  44.590000
2016-01-11 17:40:00          60      40  19.89  46.333333  19.2  44.530000

                        T3       RH_3         T4       RH_4    ...      \
date                                                          ...
2016-01-11 17:00:00  19.79  44.730000  19.000000  45.566667    ...
2016-01-11 17:10:00  19.79  44.790000  19.000000  45.992500    ...
2016-01-11 17:20:00  19.79  44.933333  18.926667  45.890000    ...
2016-01-11 17:30:00  19.79  45.000000  18.890000  45.723333    ...
2016-01-11 17:40:00  19.79  45.000000  18.890000  45.530000    ...

                           T9   RH_9     T_out  Press_mm_hg  RH_out  \
date
2016-01-11 17:00:00  17.033333  45.53  6.600000        733.5    92.0
2016-01-11 17:10:00  17.066667  45.56  6.483333        733.6    92.0
2016-01-11 17:20:00  17.000000  45.50  6.366667        733.7    92.0
2016-01-11 17:30:00  17.000000  45.40  6.250000        733.8    92.0
2016-01-11 17:40:00  17.000000  45.40  6.133333        733.9    92.0

                     Windspeed  Visibility  Tdewpoint        rv1        rv2
date
2016-01-11 17:00:00   7.000000   63.000000        5.3  13.275433  13.275433
2016-01-11 17:10:00   6.666667   59.166667        5.2  18.606195  18.606195
2016-01-11 17:20:00   6.333333   55.333333        5.1  28.642668  28.642668
2016-01-11 17:30:00   6.000000   51.500000        5.0  45.410389  45.410389
2016-01-11 17:40:00   5.666667   47.666667        4.9  10.084097  10.084097
```

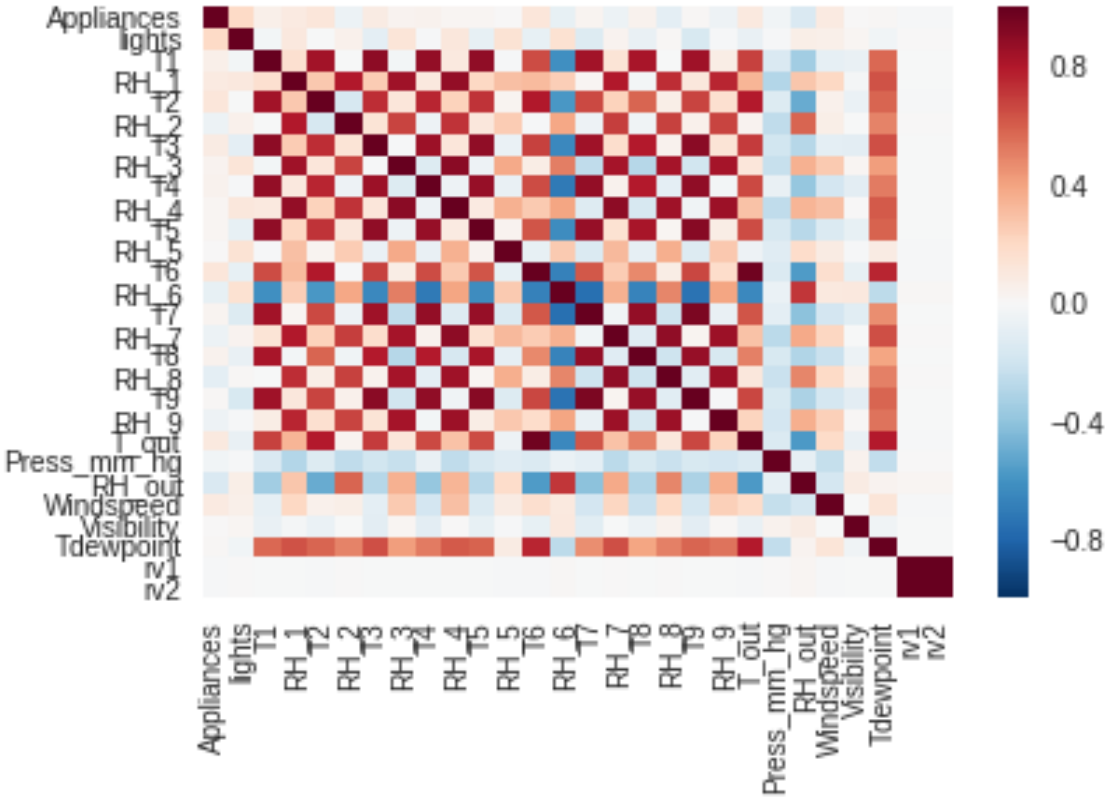
כעת נבצע מספר הדפסות לגרפים שיראו לנו איך המשתנים מתנהגים לאורך זמן. מכיוון שהדפסה של כל נתון בזמן שלו יצא צפוף מדי, סכמנו את הנתונים על פני יום אחד שלם, ואת התוצאות הצגנו בגרף:







כאן ניתן לראות את התלויות בין ה-features השונים בנתונים:



רואים בבירור שיש תלות יחסית גבוהה בין הנתונים מאותו הסוג, כלומר בין כל נתוני הטמפרטורות השונות, ובין כל הלחויות השונות – אך נבחר לא להתייחס לזה כרגע.

מהסקירה על הנתונים לא ראינו משהו משמעותי במיוחד שצריך להתייחס אליו, לכן נשאיר את הנתונים כפי שהם מופיעים מלכתחילה, ללא preprocessing.

2. Naïve Baseline Model:

בחלק זה של המשימה ניצור מודל התחלתי שיהווה בסיס להמשך. החלטנו שהמודל יחזה את אותו הערך ללא תלות בקלט, ושהערך הזה יהיה ממוצע הערכים. חשוב להדגיש שלקחנו את הממוצע של סט האימון בלבד, וחזינו את ערך זה עבור הטסט. נערוך אומדן לטיב המודל בעזרת RMSE, MSE:

```
#### Naive baseline model evaluation:
#### Train Set:
MSE: 10289.6591
RMSE: 101.4380


#### Test Set:
MSE: 11396.3106
RMSE: 106.7535
```

תוצאה זו היא טובה יחסית לממצאים באינטרנט. לצורך הדוגמא, Luis M. Candanedo et al. [1] הציגו במאמרם משנת 2016 תוצאות שנעות בין RMSE 93.21 ו-RMSE 93.18 בסט האימון והטסט בהתאמה, במודל של רגרסיה לינארית מרובה, לבין 17.56 ו-66.65 במודל של GBM.

3. Classic ML Algorithm – Gradient Boosting Machine (GBM):

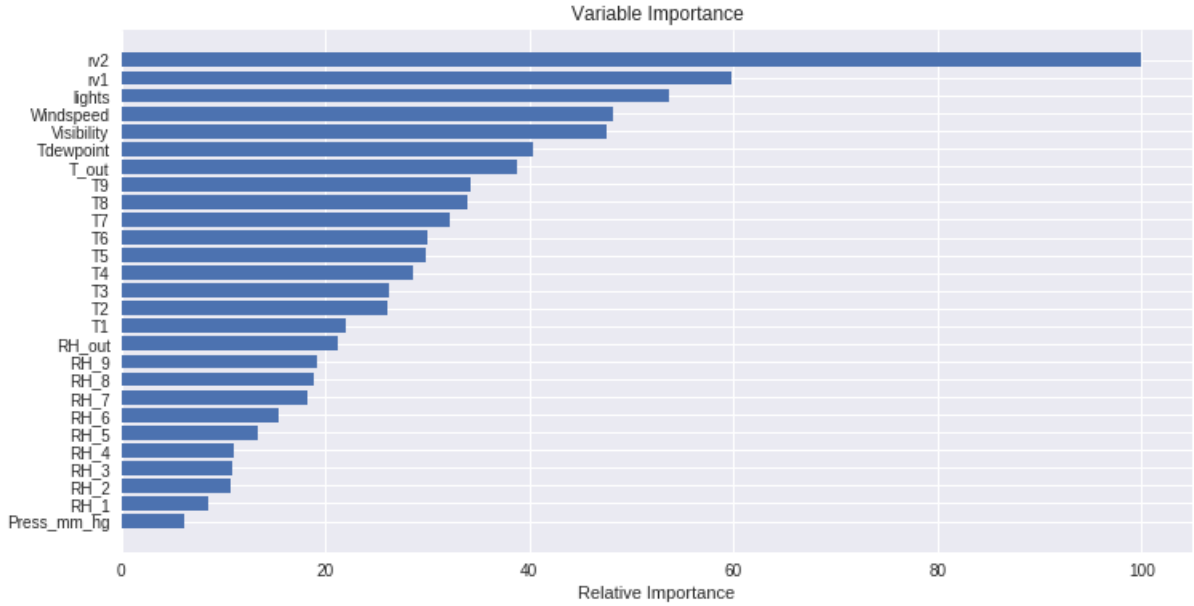
בחלק זה של המשימה בחרנו להשתמש ב-GBM מהסיבה הפשוטה ש- Luis M. Candanedo et al. [1] הגיעו למסקנה ש-GBM הציג את התוצאות הכי טובות מבין כל שאר המודלים שניסו (SVM, LR, RF).

בעזרת ההיפר-פרמטרים הבאים עבור המודל שלנו, הגענו לתוצאות יותר טובות מהמודל הנאיבי הקודם, אך שעדיין ניתנות לשיפור בעזרת מודל DNN בסעיפים הבאים.

```
params = {
    'n_estimators': 500,
    'max_depth': 4,
    'min_samples_split': 2,
    'learning_rate': 0.01,
    'loss': 'ls'}
```

```
#### Gradient Boosting Regressor
Train Set Evaluation:
MSE: 6600.9609
RMSE: 81.2463


Test Set Evaluation:
MSE: 6197.1879
RMSE: 78.7222
```

בנוסף, רצינו לראות איזה מה-features הכי תורמים למודל:

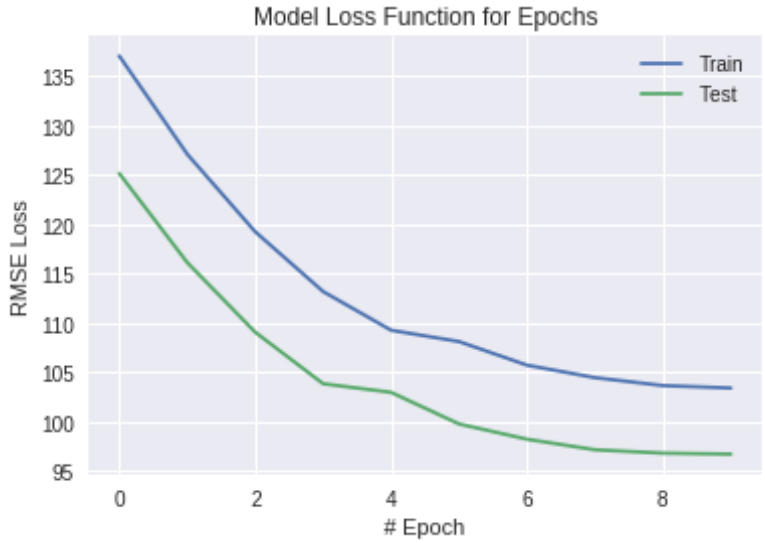


4. Deep Neural Network Model – LSTM RNN:

בתחילת השלב של בניית מודל RNN, הלכנו על מודל פשוט למדי – חיזוי אחד קדימה בעזרת 5 תצפיות אחורה (האינטרוול בנתונים הוא 10 דקות). נציע שיפורים וניישם אותם בהמשך. להלן המודל והתוצאות שלו לאחר אימון של 10 epochs, עם batch_size=10 וחילוק סט האימון מראש לסט אימון 80% וסט ולידציה 20%.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_15 (LSTM)               (None, 32)                7808
_________________________________________________________________
dense_15 (Dense)             (None, 1)                 33
=================================================================
Total params: 7,841
Trainable params: 7,841
Non-trainable params: 0
```

```
#### Train RMSE: 103.8345
#### Test RMSE: 97.2416
```



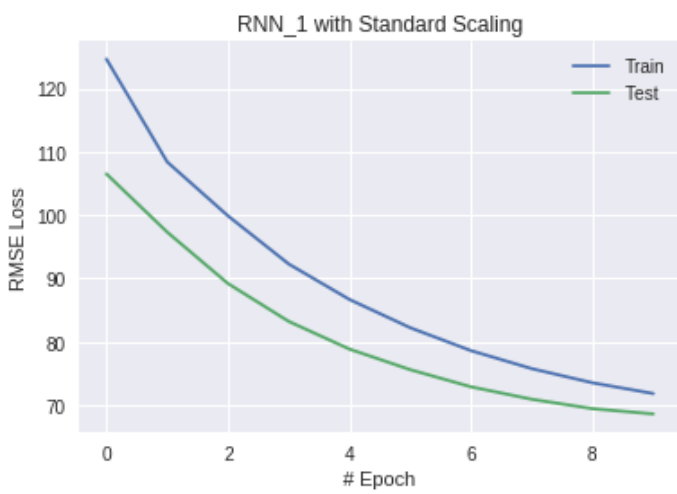
התוצאות הנ"ל לא טובות במיוחד, בעיקר כש-GBM הגיע ל-RMSE של 78.72 על הטסט, אך כנאמר זהו מודל התחלתי.

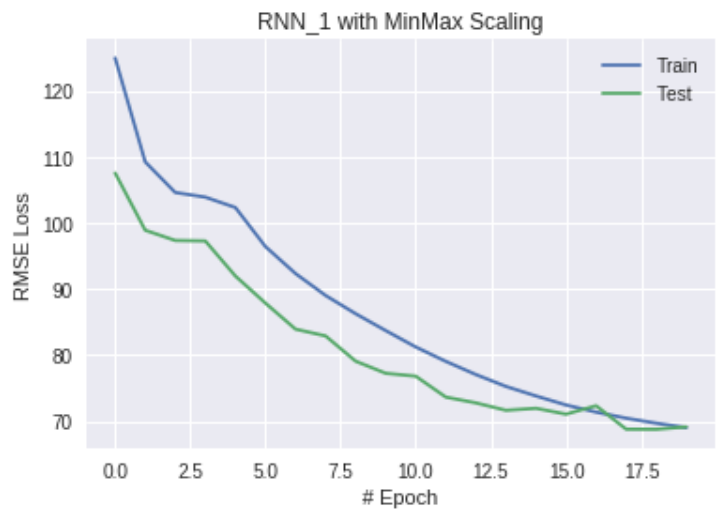
שיפורים

חשבנו על מספר שיפורים שניתן לבצע על המודל בכדי שנגיע לתוצאות טובות לפחות כמו ה-GBM ואף יותר. השיפור הראשון הוא נרמול הנתונים כך שסטיית התקן תהיה 0 – מה שיכול לעזור במיוחד ל-LSTM. הנרמול נעשה בנפרד לסט האימון וסט הבדיקה, כדי לא ליצור בעיית fitting למודל. בדקנו 2 שיטות נרמול – MinMax ו-Standard, והגענו לתוצאות מעט טובות יותר עם ה-Standard, לכן נמשיך איתו בשלבים הבאים. בנוסף, העלנו את מספר ה-epochs ל-20 כי ראינו שהמודל ממשיך ללמוד גם לאחר ה-epoch העשירי, בניגוד למודל ההתחלתי, אך בכל זאת אנחנו בהגבלת זמן ולכן נעצור ב-20. אלו התוצאות לאחר היישום:

```
#### RNN_1 (w/ MinMax scaling):
#### Train RMSE: 68.0130
#### Test RMSE: 68.5434
```

```
#### RNN_1 (w/ Standard scaling):

#### Train RMSE: 64.7250
#### Test RMSE: 65.5055
```





ניתן לראות כי התוצאות הנ"ל טובות מאוד ביחס למודל ההתחלתי ללא ה-Scaling. חשוב לציין שהנרמול נעשה רק על הנתונים ולא על ה-target, מכיוון שה-target הוא בתחום יחסית קטן ולא ראינו צורך ממשי בלנרמל גם אותו.

השיפורים הבאים הם Hyperparameter Tuning בשילוב של שינוי הארכיטקטורה של הרשת.
ניסינו מספר ארכיטקטורות שונות בשילוב עם היפר-פרמטרים שונים והגענו לתוצאות אידאליות, מעט טובות יותר מהקודמות. העלנו את גודל ה-LSTM nodes בשכבה הראשונה ל-64 והוספנו עוד שכבת LSTM, כשביניהן יש שכבות Dropout ו-BatchNormalization. בנוסף לארכיטקטורה, שינינו את ה-look_back לשעתיים אחורה (12 תצפיות אחורה), מה שהגדיל מעט את זמן הריצה הכולל של המודל. בדוגמאות הקודמות הרצנו עם batch_size=10, העלנו ל-32.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_15 (LSTM)               (None, 12, 64)            23808
_________________________________________________________________
batch_normalization_1 (Batch (None, 12, 64)            256
_________________________________________________________________
dropout_2 (Dropout)          (None, 12, 64)            0
_________________________________________________________________
lstm_16 (LSTM)               (None, 64)                33024
_________________________________________________________________
batch_normalization_2 (Batch (None, 64)                256
_________________________________________________________________
dropout_3 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_11 (Dense)             (None, 1)                 65
=================================================================
Total params: 57,409
Trainable params: 57,153
Non-trainable params: 256
_________________________________________________________________
```

```
#### Train RMSE: 55.6345
#### Test RMSE: 63.9429
```