

למידה עמוקה עבודה 2

1. בחרנו את סט הנתונים "Beijing PM2.5 Data Set" המכיל את נתוני מזג אוויר ואת רמת זיהום האוויר בביג'ין בין השנים 2010-2015 כלומר, דיווח של 5 שנים החל מתחילת 2010 ועד סוף 2014 כאשר יש דיווח כל שעה בכל יום. מדובר בבעיית רגרסיה בכך שעלינו לחזות בהתבסס על נתוני עבר את זיהום האוויר בשעה הבאה.
- ה data set מכיל 43824 רשומות כך שלכל רשומה יש 12 פיצ'רים: הפיצ'רים קלים להבנה ומתייחסים לנתוני מזג האוויר:

No: row number
year: year of data in this row
month: month of data in this row
day: day of data in this row
hour: hour of data in this row
pm2.5: PM2.5 concentration
DEWP: Dew Point
TEMP: Temperature
PRES: Pressure
cbwd: Combined wind direction
lws: Cumulated wind speed
ls: Cumulated hours of snow
lr: Cumulated hours of rain

ערכי זיהום אפשריים (ערכי target אפשריים): 0-994

ערכי הזיהום הנפוצים ביותר:

16.0	626
11.0	596
13.0	589
12.0	578
17.0	572

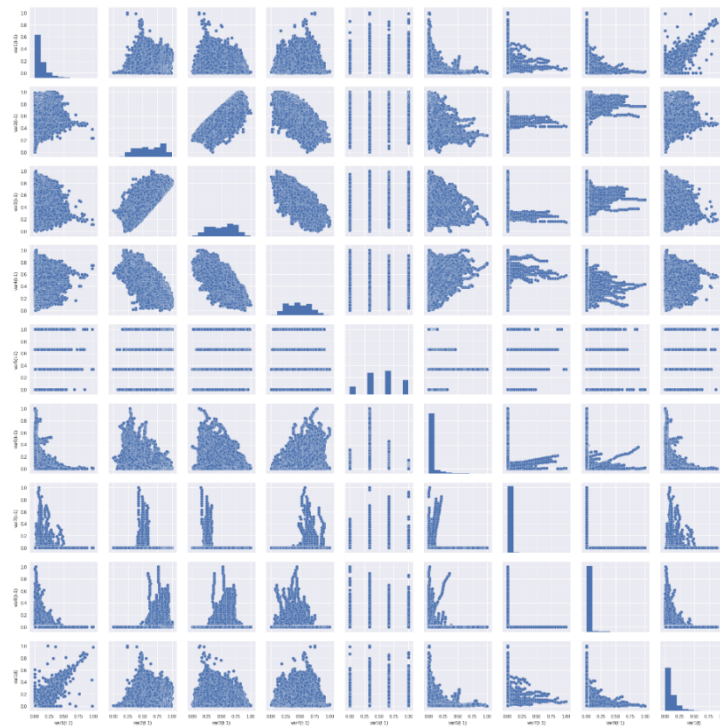
מצאנו מאמר ובלוג על ה data set הנ"ל עם שימוש בעיקר ב-LSTM:

- https://www.ischool.berkeley.edu/sites/default/files/sproject_attachments/deep-air-forecasting_final.pdf
- <https://machinelearningmastery.com/multivariate-time-series-forecasting-lstms-keras/>

יש לציין שהבדיקות על הנתונים בוצעו לאחר הסרת הפיצ'ר 'No' המתאר את אינדקס הרשומה, ולאחר שהפכנו את כל הפיצ'רים שמתארים את תאריך הרשומה לאינדקס הרשומה (כפי שמתואר בסעיף 2).

רוני מינדלין מילר 302242870
מיה קרמר 204219976

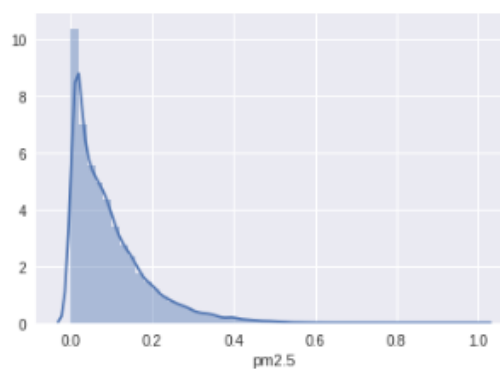
בדיקת קורלציה בין כל זוג פיצ'רים:



	pm2.5	DEWP	TEMP	PRES	cbwd	lws	Is	Ir	pm2.5_next_day
pm2.5	1.000000	0.157559	-0.090826	-0.045511	0.187448	-0.234327	0.022225	-0.049048	0.955350
DEWP	0.157559	1.000000	0.824425	-0.778723	0.232942	-0.296635	-0.034487	0.125051	0.155297
TEMP	-0.090826	0.824425	1.000000	-0.827199	0.175610	-0.154811	-0.092730	0.049034	-0.090737
PRES	-0.045511	-0.778723	-0.827199	1.000000	-0.168965	0.185253	0.069036	-0.079837	-0.043935
cbwd	0.187448	0.232942	0.175610	-0.168965	1.000000	-0.200006	0.010355	-0.048326	0.207142
lws	-0.234327	-0.296635	-0.154811	0.185253	-0.200006	1.000000	0.021889	-0.010125	-0.234346
Is	0.022225	-0.034487	-0.092730	0.069036	0.010355	0.021889	1.000000	-0.009553	0.023343
Ir	-0.049048	0.125051	0.049034	-0.079837	-0.048326	-0.010125	-0.009553	1.000000	-0.054255
pm2.5_next_day	0.955350	0.155297	-0.090737	-0.043935	0.207142	-0.234346	0.023343	-0.054255	1.000000

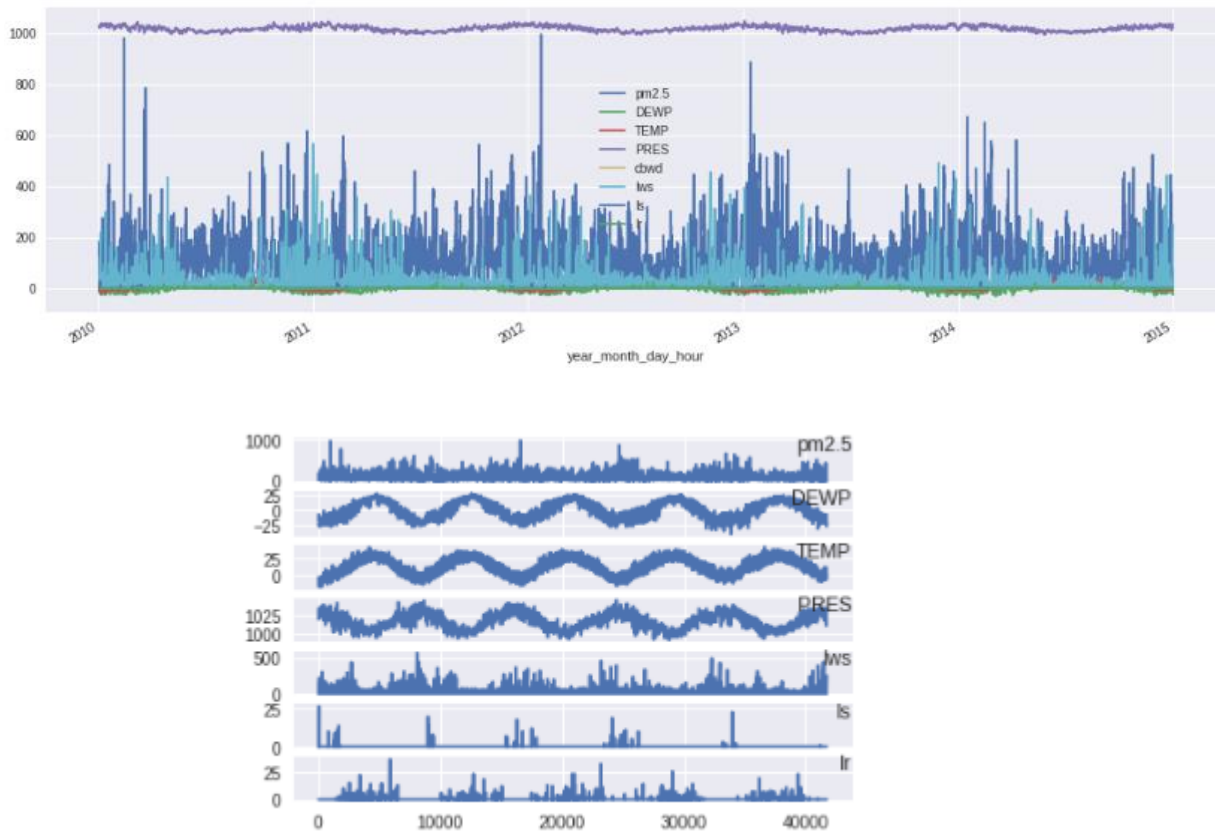
ניתן לראות שהקורלציה הכי גבוהה היא בין ערך הזיהום ביום t לבין ערך הזיהום ביום $t-1$.

התפלגות הזיהום:



רוני מינדלין מילר 302242870
מיה קרמר 204219976

התנהגות הנתונים בזמן:



ניתן לראות שאכן הנתונים מתנהגים בצורה מחזורית ולכן נסיק שהם תלויים בזמן, כלומר לתאריך יש השפעה על ערכי הפיצ'רים ולכן בזמנים שונים נצפה לערכים שונים בנתונים. נצפה שעובדה זאת תעזור לחיזוי ערך הזיהום.

2. ראשית נתאר את שלבי הpreprocess שביצענו על מנת להשתמש בdata set:
- לאחר מעבר על הנתונים זיהינו שיש ערכים חסרים בעמודת הזיהום (pm2.5) ב-24 השעות הראשונות ולכן נסיר את 24 הרשומות הנ"ל. בנוסף קיימים ערכים חסרים- תחילה נחליף ערכים חסרים ב-0 (החלפה ב-0 רק לשם הנוחות כרגע, בהמשך שנינו זאת). העמודה היחידה בה קיימים ערכים חסרים היא עמודת ערך הזיהום (pm2.5):

רוני מינדלין מילר 302242870
מיה קרמר 204219976

```
data.isna().sum()
No      0
year    0
month   0
day      0
hour     0
pm2.5   2043
DEWP    0
TEMP    0
PRES    0
cbwd     0
Iws     0
Is       0
Ir       0
```

את כל הפיצ'רים המתארים את תאריך הרשומה נאחד לפיצ'ר אחד של תאריך המתאר את השעה, יום, חודש ושנה אותה הרשומה מתארת. נעדכן את האינדקס של data frame להיות פיצ'ר זה מאחר ואלו נתונים התלויים בזמן.

בנוסף נסיר את עמודת "No" מאחר והיא מציינת את אינדקס השורה ולכן מיותרת.
כעת ננרמל את הנתונים:

מלבד הפיצ'ר המתאר את כיוון הרוח (cbwd), כל שאר הפיצ'רים מספריים.

עבור cbwd נשתמש בlabel encoder על מנת להחליף כל קטגוריה הקיימת בפיצ'ר זה במספר מתאים: [0, 1, 2, 3] 'SE', 'CV', 'NW', 'NE']

כעת ננרמל את כל הפיצ'רים (כולל תוצאת ה-label encoder על cbwd) לערכים בין 0 ל-1 ע"י שימוש ב-MinMaxScaler

כעת נעביר את הנתונים למבנה המתאים לנתונים בזמן כך שיותר נוח לראות לכל רשומה באיזה פיצ'רים וערכים נשתמש כדי לחזות את הערך שלה. ראשית ננסה לחזות את הזיהום **בשעה הנוכחית** (t) על בסיס הנתונים **מהשעה הקודמת** (t-1) ולכן לכל רשומה נוסיף את ערך הזיהום של הרשומה הבאה וכך נקבל שלכל רשומה הערך שאותו נרצה לחזות הוא ערך הזיהום של השעה הבאה. כלומר במבנה זה הפיצ'רים הנכנסים למודל הלומד הם כל 8 הפיצ'רים הראשונים, ומשתנה המטרה הוא הפיצ'ר האחרון הנקרא var1(t).

הקוד עליו הסתמכנו בבניית מבנה זה מתואר בבלוג הבא:

<https://machinelearningmastery.com/convert-time-series-supervised-learning-problem-python/>

	var1(t-1)	var2(t-1)	var3(t-1)	var4(t-1)	var5(t-1)	var6(t-1)	var7(t-1)	var8(t-1)	var1(t)
year_month_day_hour									
2010-01-02 01:00:00	0.379248	-1.235575	-1.349834	0.345882	0.358713	-0.441885	-0.069371	-0.137704	0.585205
2010-01-02 02:00:00	0.585205	-1.166272	-1.349834	0.345882	0.358713	-0.424093	-0.069371	-0.137704	0.704443
2010-01-02 03:00:00	0.704443	-0.889059	-1.431845	0.443239	0.358713	-0.406301	-0.069371	-0.137704	0.942920
2010-01-02 04:00:00	0.942920	-0.611845	-1.431845	0.540597	0.358713	-0.370518	1.245411	-0.137704	0.476806
2010-01-02 05:00:00	0.476806	-0.611845	-1.431845	0.540597	0.358713	-0.352726	2.560194	-0.137704	0.162451

רוני מינדלין מילר 302242870
מיה קרמר 204219976

א. Validation strategy - בהתחלה החלוקה ל train set ו- test set בוצעה כך שה- train set מכיל את 4- השנים הראשונות: מתאריך 1.1.2010 ועד לתאריך 31.12.2013. ה- test set יכיל את נתוני השנה האחרונה: מתאריך 1.1.2014 ועד לתאריך 31.12.2014. קיבלנו שה- train set מכיל 35040 רשומות, וה- test set מכיל 8759 רשומות כלומר ה- train set מהווה כבערך 80% מכלל הנתונים וה- test set מהווה כבערך 20% מכלל הנתונים.

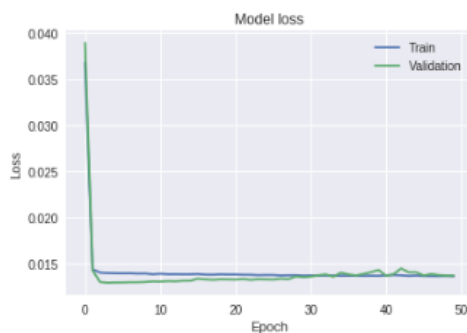
בהמשך נראה שהחלוקה השתנתה (עדיין חלוקה ל train ו- test אך בצורה שונה) ועבור מודל LSTM חילקנו את ה- train גם ל train ו- validation כפי שמתואר בהמשך.

ב. נבנה מודל בסיסי שישמש אותנו כ- baselines לשאר המודלים שנבחנו. המודל הבסיסי יחזה את ערך הזיהום על בסיס ממוצע ערכי הזיהום ב- **3 השעות האחרונות**. נבדוק את דיוק המודל באמצעות חישוב RMSE קיבלנו: TRAIN RMSE=34.352, TEST RMSE=30.928.

ג. בנינו מודל ML קלאסי מסוג XGBRegressor. המודל בונה 100 עצים שונים כך שכל עץ "משתפר" על בסיס הטעות בעץ הקודם שנבנה כאשר ה- learning rate שקבענו הוא 0.08 והעומק המקסימלי של כל עץ הוא 4. באמצעות מודל זה קיבלנו RMSE = 23.837 על ה- test set.

ד. כעת נבנה מודל NN לחיזוי ערך הזיהום בשעה הנוכחית על בסיס הנתונים מהשעה הקודמת עם שימוש בשכבת LSTM המתאימה ל- time series:
ראשית נבנה מודל פשוט שבשכבה הראשונה LSTM עם 50 ניוונים ובשכבה האחרונה (output) ניוון בודד. פונקציית ההפסד תהיה מבוססת על MAE וה- optimizer יהיה מסוג adam.
test set-על RMSE=25.3307
train set-על RMSE=28.969

פונקציית ההפסד מול מספר ה- epochs:



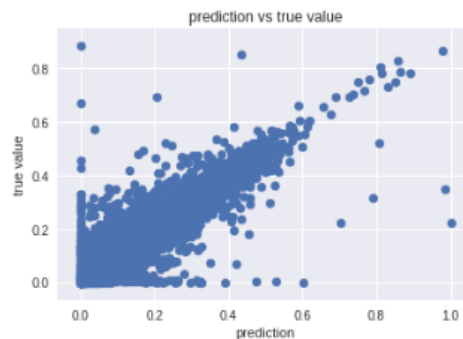
חיזוי מול הערך האמיתי test set:



רוני מינדלין מילר 302242870
מיה קרמר 204219976

ניתן לראות שהתוצאה יחסית טובה- החיזוי אכן קרוב לערך האמיתי של הזיהום וניתן לראות זאת על פי הצורה הלינארית שהתקבלה ביחס בין החיזוי לערך האמיתי.

חיזוי מול ערך אמיתי train set:



ניתן לראות שקיבלנו צורה פחות לינארית מאשר עבור plot שהתקבל עבור חיזוי מול ערך אמיתי של test set. תוצאה זאת תואמת גם את תוצאת ה-RMSE על תוצאות המודל על ה-train set.

חמשת הרשומות שקיבלו חיזוי בעל טעות מינימלית:

	pm2.5	DEWP	TEMP	PRES	cbwd	Iws	Is	Ir	true_next_day_pm2.5	pred_next_day_pm2.5	true_norm_val	pred_norm_val	dist
year_month_day_hour													
2014-04-30 04:00:00	97.0	9.0	13.0	1010.0	0.0	0.89	0.0	0.0	100.0	99.996375	0.100604	0.100600	0.000004
2014-01-12 02:00:00	0.0	-24.0	-1.0	1034.0	1.0	24.14	0.0	0.0	0.0	-0.004087	0.000000	-0.000004	0.000004
2014-01-17 01:00:00	13.0	-8.0	6.0	1030.0	1.0	101.92	0.0	0.0	12.0	11.994442	0.012072	0.012067	0.000006
2014-07-01 04:00:00	76.0	17.0	24.0	1005.0	2.0	55.43	0.0	0.0	86.0	86.006384	0.086519	0.086526	0.000006
2014-04-25 16:00:00	90.0	7.0	26.0	1014.0	2.0	11.17	0.0	0.0	96.0	96.006725	0.096579	0.096586	0.000007

חמשת הרשומות שקיבלו חיזוי בעל טעות מקסימלית:

	pm2.5	DEWP	TEMP	PRES	cbwd	Iws	Is	Ir	true_next_day_pm2.5	pred_next_day_pm2.5	true_norm_val	pred_norm_val	dist
year_month_day_hour													
2014-01-31 00:00:00	137.0	-7.0	-1.0	1021.0	2.0	39.79	0.0	0.0	469.0	141.355458	0.471831	0.142209	0.329622
2014-04-09 20:00:00	580.0	-3.0	22.0	1015.0	2.0	12.96	0.0	0.0	217.0	556.024189	0.218310	0.559380	0.341071
2014-12-09 11:00:00	0.0	-8.0	-3.0	1036.0	0.0	1.79	0.0	0.0	339.0	-2.772151	0.341046	-0.002789	0.343835
2014-11-20 20:00:00	375.0	0.0	3.0	1018.0	3.0	0.89	0.0	0.0	0.0	387.927101	0.000000	0.390269	0.390269
2014-04-09 19:00:00	80.0	-1.0	24.0	1013.0	2.0	7.15	0.0	0.0	580.0	84.208600	0.583501	0.084717	0.498784

ניתן לראות שהמודל תמיד חוזה (t) ערך יחסית קרוב לערך של הזיהום ביום הקודם (t-1). כאשר אכן הזיהום ביום t קרוב לזיהום ביום t-1 אז ניתן לראות שהמודל חוזה בצורה טובה (כמו שרואים בחמשת הרשומות שקיבלו חיזוי בעל טעות מינימלית). לעומת זאת כאשר הזיהום ביום t שונה מאוד מהזיהום ביום t-1 אז המודל לא מצליח לבצע חיזוי בצורה טובה (כמו שרואים בחמשת הרשומות שקיבלו חיזוי בעל טעות מקסימלית). מכאן אנו מסיקות 2 דברים:

1. הסרה או מילוי ערכים חסרים בדרכ אחרת ולא ע"י 0.
2. יש לנסות לחזות את הזיהום ביום t ע"י יותר מיום אחד אחורה (t-1).

רוני מינדלין מילר 302242870
מיה קרמר 204219976

ה. מהמסקנות שהצגנו לעיל חשבנו על מספר פתרונות לשיפור המודל:

- ✓ הסרת ערכים חסרים- מאחר ויש 2067 רשומות בעלי ערך pm2.5 חסר מתוך 43824 רשומות החלטנו לנסות להסיר את הערכים החסרים ולא להשלים אותם ע"י אפסים.
- ✓ הגדלת מספר הרשומות שבאמצעותן אנו מבצעות חיזוי לרשומה הנוכחית מ-1 ל-48.
- ✓ שימוש עבור המשתמש הקטיגוריאלי cbwd המתאר את כיוון הרוח ב-one hot encoding ע"י שימוש ב-get dummies (הוספת עמודה עבור כל ערך אפשרי ואתחולן ב-0, כך שרק בעמודה בעלת הערך המתאים לרשומה יהיה 1) במקום שימוש ב-label encoder ונרמול הערך הזה כי בצורה הזאת אנו יוצרות קשר של גדול/קטן בין הערכים השונים- דבר שאינו משקף תכונה אמיתית שלא פיצ'ר זה.
- ✓ שינוי מבנה הרשת- הוספת שכבת LSTM ו-Batch Normalization.

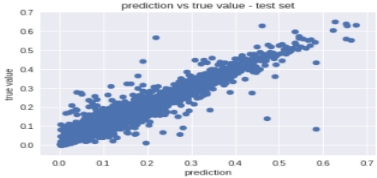
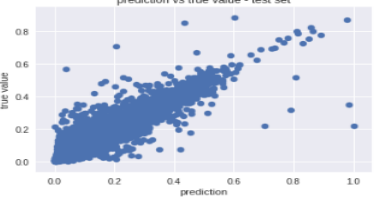
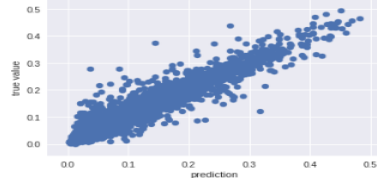
הערה: המסקנות כבר נכתבו בסדר ממוין לפני חשיבותן (נממש את כולן כי לדעתנו כל הפתרונות שכתבנו עשויים לשפר את התוצאות של המודל).

יש לציין שכל שיפור שנבצע יהיה על בסיס השיפורים הקודמים שביצענו קודם במידה ואכן יהוו שיפורים למודל (כלומר אם בשיפור הראשון הורדנו ערכים חסרים ואכן קיבלנו RMSE טוב יותר, אז בשאר השיפורים שנבצע הdata יהיה ללא ערכים חסרים וכו').

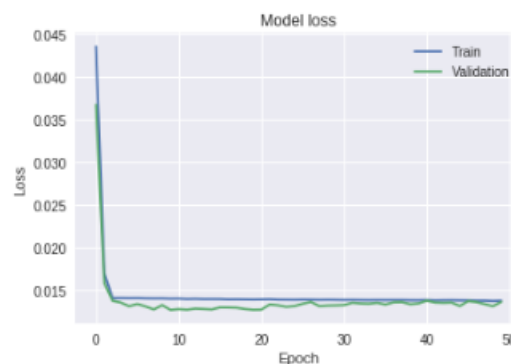
• הסרת ערכים חסרים:

לאחר הסרת הערכים החסרים החלטנו לשנות גם את החלוקה לtrain ו-test שתהיה יותר הוגנת מאחר וב-test נותרו רק 0.16% מכלל התצפיות- חילקנו את הנתונים בצורה כזאת שבtrain יהיו 0.8 מהתצפיות וב-test יהיו את 0.2 התצפיות הנותרות כאשר החלוקה אינה רנדומלית אלא לפי תאריך (בtrain התקבלו 0.8 מהתצפיות הראשונות). את הtrain חילקנו לאחר מכן לtrain ו-val באותה צורה.

תוצאות:

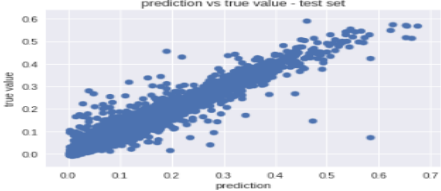
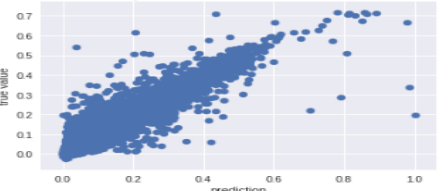
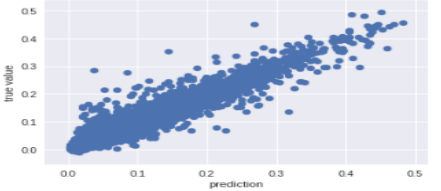
Data set	RMSE	
test	23.199	
train	26.79	
validation	21.60	

ניתן לראות שאכן יש שיפור משמעותי מאשר התוצאות הקודמות.

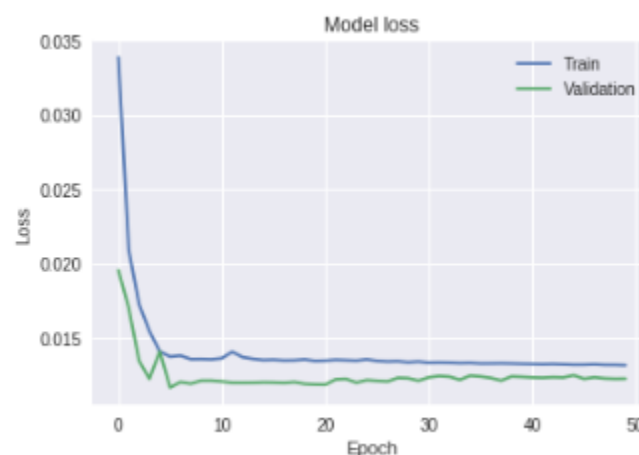


רוני מינדלין מילר 302242870
מיה קרמר 204219976

- הגדלת מספר הרשומות שבאמצעותן אנו מבצעות חיזוי לרשומה הנוכחית מ-1 ל-48.
תוצאות:

Data set	RMSE	
test	21.54	
train	24.82	
validation	19.56	

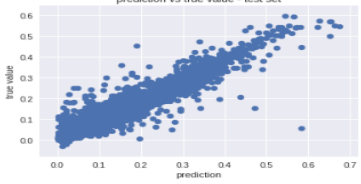


ניתן לראות שקיבלנו תוצאות טובות יותר בצורה משמעותית ולכן נמשיך לחזות את הזיהום בשעה הבאה על פי נתוני 48 השעות הקודמות.



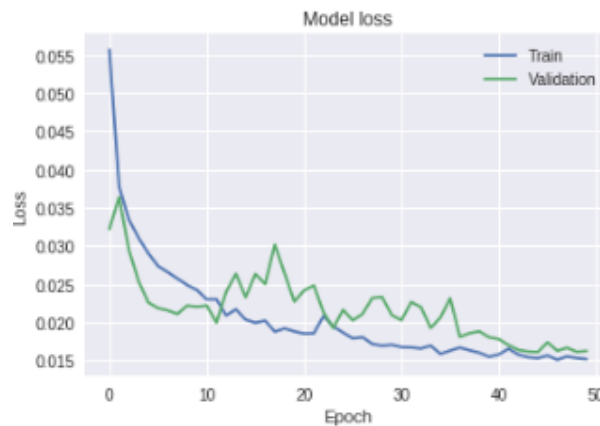
הערה: notebook מבוססת על פתרון זה (זהו הפתרון הטוב ביותר שמצאנו לנתונים) ולכן יש בה עוד מספר גרפים שלא צירפנו לדוח על ההתאמה בין החיזוי לערך האמיתי של ה-train, test, validation sets.

רוני מינדלין מילר 302242870
מיה קרמר 204219976

- שימוש עבור המשתנה הקטגוריאל cbwd המתאר את כיוון הרוח ב-one hot encoding תוצאות:

Data set	RMSE	
test	26.33	
train	26.63	
validation	23.92	

ניתן לראות שקיבלנו תוצאות פחות טובות בצורה משמעותית מכל שאר התוצאות קודם.

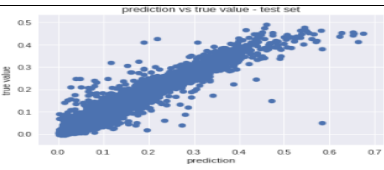
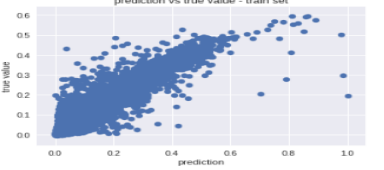
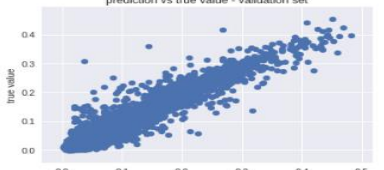


מאחר וחשבנו שיש בעיה עם ההתכנסות של המודל בעקבות הוספת ה-one hot encoding, ניסנו להקטין את ה- lr של ה-optimizer, אך פתרון זה לא שיפר את התוצאות ואף חזרו תוצאות יותר גרועות ולכן החלטנו להחזיר את הפיצ'ר לנרמול הראשוני שביצענו (שינוי הערכים הקטגוריאלים למספריים ע"י label encoder ונרמול למספר בין 0 ל-1).

רוני מינדלין מילר 302242870
מיה קרמר 204219976

✓ שינוי מבנה הרשת- הוספת שכבת LSTM ו-Batch Normalization

- הוספת שכבת Batch Normalization מורידה את דיוק המודל בחיזוי- קראנו על זה וגילינו ש-Batch Normalization עשוי לגרום לבלגן בין time stamp.
- הוספת שכבת LSTM נוספת בעלת 50 ניוונים גם הורידה את דיוק המודל:

Data set	RMSE	
test	27	
train	29.16	
validation	23.86	

לסיכום- קיבלנו את התוצאות הטובות ביותר לאחר הסרת ערכים חסרים ושימוש ב48 השעות האחרונות לחיזוי הזיהום בשעה הנכחית.