

פרוייקט מסכם בלמידה חישובית יישומית

Self-training with Noisy Student improves ImageNet classification

Reference: Xie, Q., Luong, M. T., Hovy, E., & Le, Q. V. (2020). Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 10687-10698).

קישור לגיט המכיל את קוד הפרויקט:

<https://github.com/shaharbsh/ML-final-project.git>

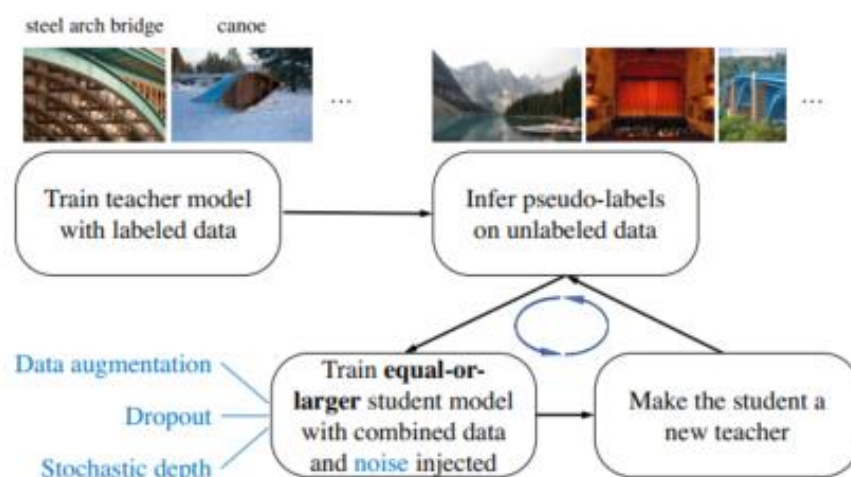
תקציר:

המאמר מציג אלגוריתם המבצע semi-supervised learning אשר משתמש במידע לא מתויג על מנת לשפר את ביצועי המודלים המאומנים רק על מידע מתויג. אלגוריתם זה הצליח לשפר את מודל ה-SOTA בכ- 2% במדד ה-accuracy, ולהראות שיפור קונסיסטנטי עבור כל מודל שהוכנס לאלגוריתם.

שלב 1 – בחירת אלגוריתם:

הסבר על האלגוריתם מהמאמר:

המאמר מציג אלגוריתם שמטרתו לשפר את הביצועים של מודל קיים על ידי אימון על מידע לא מתויג הנלקח מהאינטרנט. במאמר הם משפרים את מודל EfficientNet אשר אומן במקור על מידע מ-image net (המכיל קרוב ל- 15 מיליון תמונות מתויגות עם למעלה מ- 20 אלף סיווגים) על ידי שימוש במידע של 300 מיליון תמונות ללא תיוג השייכות לסיווגים הרלוונטיים שנלקחו מתוך האינטרנט. האלגוריתם המוצע הינו אלגוריתם איטרטיבי הפועל בצורה הבאה (התמונה לקוחה מהמאמר):



1. השלב הראשון הינו אימון ה"מורה": בשלב זה מאמנים את המודל על המידע המתויג על מנת לקבל מודל מאומן.
2. השלב השני הינו אימון ה"תלמיד": שלב זה הינו השלב האיטרטיבי כאשר הוא מחולק לכמה תתי שלבים:
 - a. תת שלב א: יצירת pseudo labels
 - i. יצירת תיוגים לחלק מהמידע הלא מתויג
 - ii. העברת threshold על סבירות התיוג על מנת להשאיר רק תיוגים חזקים (זהו hyper parameter של האלגוריתם)
 - iii. ביצוע hard prediction - קבלת תיוג יחיד לכל תמונה על בסיס ההסתברות הגבוה ביותר שהמודל נתן ($\arg \max$)
 - iv. איזון המידע שנשאר על מנת לקבל מידע מאוזן לאימון ה"תלמיד"
 - b. תת שלב ב: אימון התלמיד על בסיס ה- pseudo labels (בנוסף למידע המקורי), יחד עם הוספת אוגמנטציות, dropout ועומק סטוכסטי.
 - c. תת שלב ג: הפיכת ה"תלמיד" ל"מורה" וסיום האיטרציה הנוכחית (כאשר ניתן להמשיך בתהליך לפי כמות האיטרציות שבחרים- זהו hyper parameter של האלגוריתם).

Pseudo Code:

```

Teacher data ← Get image net data
unlabeld data ← Get unlabeld data
teacher ← efficientNet.fit(Teacher data )
For i in range(num_of_students):
    Create pseudo labels:
        Pred teacher ← teacher.predict(unlabeld data)
        Pseudo labels ← argmax(Pred teacher.max_vlaue > threshold)
        Pseudo labels ← balance_data(Pseudo labels)
    student data ← image net data + (unlabeld data, Pseudo labels)
    student ← efficientNet.fit(student data, augment=true, dropout=true,
    stochastic_depth=true)
    Teacher ← student
  
```

המודל עליו מבוסס האלגוריתם במאמר הוא EfficientNet עם dataset המכיל 300 מיליון תמונות לא מתויגות ועוד כ- 15 מיליון תמונות מתויגות. אין לנו כוח עיבוד או זיכרון (או זמן) על מנת להתמודד עם גודל דאטה כזה ומודל כה עמוק הדורש כמויות כאלו של מידע. (גם במאמר ישנם ניסויים אשר הם ביצעו על מודל

חלקי - הרבה פחות עמוק, משום שלמרות שיש להם משאבים של גוגל הם כתבו שאין להם מספיק כוח עיבוד כדי להריץ את כל הניסויים).

לכן (בתיאום עם המרצה) השתמשנו ב**מודל CNN פשוט** בתור baseline (המתואר בשלב 3) ועליו בנינו את אלגוריתם ה-noisy student שהמאמר מציע והראנו איך האלגוריתם ה-noisy student המתואר במאמר אכן משפר את מודל ה-baseline.

שינויים שביצענו בעקבות שינוי הרשת וגודל המידע:

- **teacher factor:** במאמר השתמשו בכל המידע המתויג שיש על מנת לאמן את ה"מורה" הראשון. אצלנו משום שכל המידע מתויג נאלצנו לקבוע teacher factor שיקבע את אחוז התמונות שניקח מה-dataset על מנת לאמן את מודל ה"מורה". השתמשנו ביתר המידע המתויג כמידע לא מתויג (לא השתמשנו בתיוג של תמונות אלו). הערך שבחרנו עבור פרמטר זה הוא 0.1, כלומר 10% מהתמונות ב-dataset. ב-datasets גדולים מאוד, הקטנו את ערך הפרמטר ל-5% מהתמונות שיש ב-dataset על מנת לקצר את זמן ריצת האלגוריתם.
- **מספר ה-epochs:** במאמר הריצו את מודל ה-student עם 350 epochs ואף עם 700 epochs עבור student קטן יותר. אצלנו, משום שיש לנו הרבה פחות דאטה, הרצנו את מודל ה-noisy student ומודל ה-baseline עם 5 epochs ואף עם 2 epochs עבור datasets קטנים יותר. סביר להניח כי הגדלת מספר ה-epochs הייתה משפרת את התוצאות אך בהתאם למגבלות כוח החישוב שיש לנו נאלצנו להסתפק במעט (ועדיין האלגוריתם משפר את תוצאות המודל ה-baseline).
- **גודל ה-batch:** במאמר גודל ה-batch הוא 2048, כאשר אצלנו יש הרבה פחות דאטה ולכן גודל ה-batch שהשתמשנו בו עבור המודלים שהרצנו הוא 32.
- **threshold:** במאמר בדקו threshold של 0.3 על מנת להחליט האם הסיווג הוא חזק ובכך לבחור האם להשאיר את התמונה עם ה-pseudo label שקיבלנו עבורה ל"תלמיד". הבחירה ב-0.3 התבצעה עבור dataset עם 300 מיליון תמונות לא מתויגות ולמעלה מ-20 אלף סיווגים. אצלנו יש datasets הרבה יותר קטנים עם לכל היותר 10 סיווגים ולכן החלטנו להגדיל את ערך ה-threshold ולבדוק מה הערך האופטימלי. זהו אחד מה-hyper params שבחרנו לבדוק עבור האלגוריתם.
- **מספר ה"תלמידים":** במאמר הם ביצעו תהליך אופטימיזציה למציאת הפרמטרים הטובים ביותר וגילו כי 3 תלמידים נותנים את הביצועים הטובים ביותר. משום שאצלנו כמות המידע קטנה בהרבה, החלטנו לבדוק את הפרמטר הזה בתהליך ה-hyperparameters optimization ובחרנו לבדוק את הפרמטר הזה עם ערכים הקרובים ל-3 כפי שנבחר במאמר.

יתרונות של האלגוריתם:

- כיום יש המון מידע לא מתויג יחסית זמין באינטרנט, האלגוריתם משתמש במידע זה על מנת לשפר את הביצועים של מודלים אשר אומנו על מידע מתויג יחסית יותר קטן (15 מיליון תמונות לעומת 300 מיליון תמונות) ובכך מאפשר לשפר את הביצועים.
- לעומת אלגוריתמים אחרים שמנסים לבצע semi-supervised learning ומצליחים רק עם שימוש ב- 3.5 ביליון תמונות, אלגוריתם זה מצליח להגיע להישגים טובים יותר עם רק 300 מיליון תמונות דבר המהווה שיפור משמעותי בכמות המידע שצריך לאסוף, בזמני הריצה ובכוח החישוב הדרושים.
- האלגוריתם הינו יחסית כללי ולכן מאפשר לקחת מודלים שונים ולפעיל אותו עליהם על מנת לשפר את התוצאות שלהם.

חסרונות של האלגוריתם:

- האלגוריתם דורש גישה לכמות מידע די גדולה שאמנם לא מתויגת אך כן עם אינדיקציה שהיא רלוונטית למשימה הדרושה. איסוף מידע זה יכול להיות לא פשוט אך משמעותית יותר קל מתיוג מסודר.
- עקב אופיו האיטרטיבי של האלגוריתם אנו מאמנים מודלים מספר רב של פעמים, דבר היכול לקחת המון זמן וכוח חישוב מאד גדול על מנת לבצע (לצורך המחשה לקח לכתובי המאמר 6 ימים עם ה-GPU הכי חזקים שיש ל-google להציע על מנת לאמן את המודל שלהם- לנו לקח 4 ימים להריץ את המודלים שלנו על מספר GPU שונים על גבי 20 datasets שונים, זאת עם שימוש בגודל dataset קטן בהרבה ומודל פשוט יותר שעליו מריצים את האלגוריתם).

שלב 2 – הצעת שיפור לאלגוריתם:

שיפור האלגוריתם:

בחרנו לשפר את האלגוריתם מהמאמר על ידי ביצוע ensemble על ה- teachers ועל ה- students. תחילה אימנו מספר teachers המבצעים את אלגוריתם ה- baseline (כאשר אחד מהם הינו מודל ה- baseline שאליו אנו משווים את התוצאות) על אותן תמונות מתוך ה- dataset. לאחר מכן ביצענו soft ensemble, כלומר ממוצע על וקטורי ההסתברויות אשר ה- teachers מוציאים בפרדיקציה שלהם, על מנת להשתמש בהם ליצירת ה- pseudo labels. לאחר מכן אימנו את אותו מספר students על פי אלגוריתם ה- noisy student על מנת לקבל ensemble של students – באיטרציה הבאה הם יהיו ה- teachers החדשים. כאשר האלגוריתם מסיים לרוץ אנו נקבל את ה- ensemble של ה- students ועל בסיסם נבצע את הפרדיקציה.

את גודל ה- ensemble בחרנו על ידי ביצוע hyperparameters optimization על הפרמטר ensemble size בין המספרים 2,3. גודל ה- ensemble משפיע משמעותית על זמן הריצה ולכן לא ניסינו לבדוק ensemble size גדול יותר מ- 3.

בחרנו בביצוע ensemble משום שישנה השפעה רבה למשקלים הרנדומליים המאותחלים עבור המודל המקורי (מודל ה-CNN), ובכך שמבצעים את המודל מספר פעמים מפחיתים את השפעת המשקלים על התוצאות ובכך מעלים את דיוק התוצאות.

קיבלנו כי ברוב ה-datasets ישנו שיפור על ידי ביצוע ה-ensemble אך לא בכל ה-datasets (ניתן לראות את התוצאות בקובץ ה-excel המצורף).

במקרה זה מוצא האלגוריתם הינו מספר מודלים של student כאשר על מנת לקבל פרדיקציה אנו משתמשים בשיטת ה-ensemble (ממוצע על הסתברויות הסיווגים לפני ביצוע arg max למציאת הסיווג), ביצענו זאת בעזרת פונקציית predict ייעודית שבנינו.

שלב 3 – בחירת אלגוריתם ידוע להשוואה:

עקב מגבלות כוח החישוב וזמן הריצה של המודל ובתיאום עם המרצה, בחרנו אלגוריתם בסיסי יותר על מנת לעבוד איתו, אלגוריתם זה הינו מודל CNN לסיווג תמונות המוצע על מנת לפתור את בעיית הסיווג ב-Fashion MNIST. מודל זה מהווה את מודל ה-baseline שלנו להשוואה, כמו כן אנו משתמשים בארכיטקטורה של מודל זה על מנת לאמן את ה-teacher וה-noisy student באלגוריתם המוצע על ידי המאמר. מודל זה בנוי במפורט בקוד להלן:

```
model = tf.keras.Sequential([
    tf.keras.layers.Conv2D(32, (3,3), padding='same', activation=tf.nn.relu,
        input_shape=(input.shape[1],input.shape[2],input.shape[3])),
    tf.keras.layers.MaxPooling2D((2, 2), strides=2),
    tf.keras.layers.Conv2D(64, (3,3), padding='same', activation=tf.nn.relu),
    tf.keras.layers.MaxPooling2D((2, 2), strides=2),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(128, activation=tf.nn.relu),
    tf.keras.layers.Dense(num_of_classes, activation=tf.nn.softmax)
])
```

כפי שניתן לראות מודל CNN זה מורכב משכבת קונבולוציה (אשר מחפשת קשרים מרחביים לוקאלים חשובים) בעלת 32 פילטרים שלאחריה מבוצעת שכבת maxPooling אשר עוזרת "לשאוב" את המידע המרחבי הלוקאלי החשוב ממוצאי הפילטרים. פעולה זו מבוצעת פעמיים כאשר מספר הפילטרים משתנה ל-64 בפעם השנייה. לאחר מכן מבצעים שכבת flatten המשטחת את המוצאים הדו ממדיים של שכבת הקונבולוציה לוקטור מממד אחד. לאחר מכן נבצע שכבת fully connected בעלת 128 ניוונים ולאחריה נבצע שכבת fully connected נוספת עבור המוצא של הרשת כאשר שכבה זו היא בגודל מספר הסיווגים (גודל זה שונה עבור כל dataset על פי מספר הסיווגים באותו dataset), עבור שכבה זו פונקציית האקטיבציה שנשתמש בה היא softmax הנותנת וקטור הסתברות של הסיווגים, משום שאלגוריתם זה מבצע משימת סיווג. שימוש במוצא זה יחד עם cross entropy loss מאפשר לרשת ללמוד את הסיווגים.

שלב 4 – הערכת ביצועי האלגוריתמים מהשלבים הקודמים:

השתמשנו ב-20 datasets הלקוחים מתוך ה-datasets המוצעים על ידי ספריית Keras ומתוך datasets מאתר Kaggle. הדאטאסטים בהם השתמשנו הם:

- Fashion MNIST
 - MNIST (אותו פיצולנו ל- 2 datasets, כאשר כל dataset מכיל 5 סיווגים)
 - CIFAR10 (אותו פיצולנו ל- 2 datasets, כאשר כל dataset מכיל 5 סיווגים)
 - CIFAR100 (אותו פיצולנו ל- 10 datasets, כאשר כל dataset מכיל 10 סיווגים)
 - A-Z Handwritten Alphabets (אותו פיצולנו ל- 5 datasets, כאשר ארבעה datasets מכילים 5 סיווגים והאחרון מכיל 6 סיווגים) – dataset זה לקוח מאתר Kaggle.
- ביצענו 10-fold cross validation עבור כל datasets, כאשר עבור כל איטרציה הרצנו 3 איטרציות על מנת לבצע hyperparameters optimization כדי לחפש את ה-hyper parameters הטובים ביותר עבור כל אלגוריתם ועבור כל dataset. למודל ה-baseline מכיוון והוא לא עובר באלגוריתם אין פרמטרים של האלגוריתם אותם יש לבדוק, ולכן עבור מודל זה לא ביצענו hyperparameters optimization.
- הפרמטרים אותם בדקנו עבור אלגוריתם ה-noisy student הם:
- `unl_factor`: זהו פרמטר הקובע את אחוז התמונות ללא תיוג מתוך ה-dataset שניקח על מנת ליצור pseudo labels על מנת לאמן student באלגוריתם. עבור כל איטרציה אנו לוקחים סט אחר של מידע שעוד לא יצרנו לו pseudo labels. כלומר פרמטר זה קובע את כמות המידע הלא מתויג שנשתמש בו בכל איטרציה. הערכים שבחרנו לבדוק עבור פרמטר זה הם: 0.05, 0.1, 0.2.
 - `threshold`: זהו פרמטר שקובע את סף רמת הביטחון שמעליו נשאיר את הרשומה עם הסיווג שנבחר עבורה על ידי מודל ה-teacher שהורץ. כלומר, נשאיר את הרשומות שקיימת עבורם לפחות הסתברות אחת לקבל סיווג כלשהו שהיא מעל ערך הסף שקבענו. הערכים שבחרנו לבדוק עבור פרמטר זה הם: 0.4, 0.5, 0.6, 0.7, 0.8. נשים לב שבדרכי ה-datasets הערכים שנבחרו כטובים ביותר הם: 0.6, 0.8.
 - `num_of_students`: זהו פרמטר הקובע את כמות ה-students שנאמן, כלומר את כמות האיטרציות שנבצע לאלגוריתם noisy student. הערכים שבחרנו לבדוק עבור פרמטר זה הם: 1, 2, 3. נשים לב שבדרכי ה-datasets הערכים שנבחרו כטובים ביותר הם: 2, 3. דבר זה לא מפתיע משום שכאשר ישנו רק student אחד, האלגוריתם אינו מבצע את כל השלבים בתהליך ה-teacher – student ולא מבצעים את השלב בו ה-student הופך להיות ה-teacher עבור student חדש, שזהו העיקר באלגוריתם זה.
- עבור האלגוריתם המשפר את ה-noisy student ביצענו הנחה שהפרמטרים הטובים ביותר עבור ה-noisy student מתוך אלו שבדקנו אכן הכי טובים גם באלגוריתם המשופר וההבדל בינם הוא רק בגודל ה-ensemble ולכן הפרמטר הנוסף אותו בדקנו עבור אלגוריתם המשפר את ה-noisy student הוא:
- `ensemble_size`: זהו הפרמטר שקובע עבור אלגוריתם improved noisy student את מספר ה-teachers ו-students אותם נאמן בכל איטרציה. הערכים שבחרנו לבדוק עבור פרמטר זה הם: 2, 3. שני הערכים 2, 3 נבחרו כטובים ביותר עבור datasets שונים ועבור אלגוריתמים שונים. בחרנו במספרים אלו ולא במספרים גדולים יותר משום שהרצה של ensemble לוקחת הרבה יותר זמן מהרצת teacher אחד ובמספרים גדולים יותר זמני הרצה גדולים מאוד.

תוצאות תהליך ה-10-fold cross validation:

במאמר מדדו את טיב האלגוריתם המוצע על פי מדד accuracy ולכן גם בביצוע ה-10-fold cross validation בחרנו במדד זה להשוואת האלגוריתמים.
על פי התוצאות ניתן לשים לב כי אלגוריתם ה-noisy student אכן משפר את התוצאות של מודל ה-baseline ב-2% במדד ה-accuracy לערך.
לפי תוצאות מדד ה-accuracy נראה כי האלגוריתם שבנינו על מנת לשפר את אלגוריתם המאמר noisy student, משפר את תוצאות האלגוריתם ברוב המקרים בעוד כ-1% ב-accuracy.
מצורף קובץ excel המכיל את תוצאות 10 האיטרציות עבור כל dataset עבור כל אלגוריתם.

שלב 5 - בדיקת מובהקות סטטיסטית של התוצאות:

Friedman test:

מבחן פרידמן מאפשר לנו לבדוק האם קיימת מובהקות סטטיסטית של התוצאות של האלגוריתמים השונים, מבחן זה מקבל את התוצאות של האלגוריתמים השונים (השתמשנו במדד ה-accuracy) מ-datasets שונים ומדרג את כלל התוצאות עבור כל dataset לפי מספר האלגוריתמים השונים שאנו משווים.
על ידי שימוש בדירוג זה ניתן לקבוע ערך המאפשר לנו לבדוק מובהקות סטטיסטית – כאשר ההסתברות לקבל ערך זה הינו קטנה מ-5% (0.05).
אחרי ביצוע מבחן פרידמן על התוצאות שלנו קיבלנו p value של: $1.096225409948948e-05$ אשר קטן מ-0.05 ולכן יש מובהקות סטטיסטית. כלומר התוצאות הן לא כך באופן רנדומאלי.

Post Hoc test:

קיבלנו ממבחן פרידמן כי יש מובהקות סטטיסטית ולכן ביצענו ניתוח Post Hoc של האלגוריתמים על מנת לראות את המובהקות הסטטיסטית בין האלגוריתמים השונים, קיבלנו את התוצאות הבאות:
(improved noisy student – 2, noisy student – 1, baseline – 0)

	0	1	2
0	1.00000	0.001000	0.005830
1	0.00100	1.000000	0.236522
2	0.00583	0.236522	1.000000

מהטבלה ניתן לשים לב כי יש מובהקות סטטיסטית בין כל 2 אלגוריתמים אשר מקיימים כי הערך של ה-p value שלהם קטן מ-0.05.
במקרה זה יש מובהקות סטטיסטית בין אלגוריתם 0 (baseline) עם אלגוריתם 1 (noisy student) וגם בין אלגוריתם 0 (baseline) עם אלגוריתם 2 (improved noisy student).
אין מובהקות סטטיסטית בין אלגוריתם 1 (noisy student) עם אלגוריתם 2 (improved noisy student) מכיוון וה- $p \text{ value} = 0.23$ (גדול מ-0.05).

למרות כי התוצאות של אלגוריתם 2 ב- accuracy לרוב טובות יותר מאשר התוצאות של אלגוריתם 1, אין מובהקות סטטיסטית והאלגוריתם לא בהכרח טוב יותר מהאלגוריתם המוצג במאמר. תוצאות אלו יכולות להיות עקב הנחה שביצענו והיא כי הפרמטרים הטובים ביותר עבור ה- noisy student מתוך אלו שבדקנו אכן הכי טובים גם באלגוריתם המשופר וההבדל בינם צריך להיות רק בגודל ה- ensemble. אם היה לנו עוד זמן וכוח חישוב היינו מריצים מחדש את תהליך ה- 10-fold cross validation ומחפשים את כל הפרמטרים מחדש עבור האלגוריתם המשופר (מכיוון ולקח לנו 4 ימים של הרצת מודלים על מספר GPU שונים, אין לנו את זמן הדרוש על מנת לבצע זאת שוב), כאשר ביצענו זאת באופן נקודתי ראינו כי אכן יש שיפור מהותי (לרוב אף ב- 2% יותר טוב מ- noisy student) אך ללא בדיקה מחודשת עבור כל ה- datasets לא ניתן להראות מובהקות סטטיסטית.

סיכום ומסקנות:

הרצנו מודל CNN בתור baseline להשוות אליו את ביצועי האלגוריתם המוצע במאמר. לאחר מכן ביצענו את האלגוריתם שמוצע במאמר על אותו מודל CNN על 20 datasets שונים וראינו כי אכן האלגוריתם noisy student המוצע במאמר משפר בכ- 2% את תוצאות מודל ה- baseline אליו השונו. לאחר מכן ביצענו שיפור לאלגוריתם ה- noisy student על ידי ביצוע ensemble על כמות ה"מורים" וה"תלמידים" וראינו כי אלגוריתם זה לרוב משפר בעוד כ- 1% את תוצאות האלגוריתם מהמאמר. בעזרת 10-fold cross validation ביצענו hyperparameters optimization על מנת למצוא את הפרמטרים הטובים ביותר עבור האלגוריתמים שהרצנו ועבור ה- datasets שטענו. לאחר מכן ביצענו Friedman test על מנת לבדוק מובהקות סטטיסטית וקיבלנו כי אכן יש מובהקות סטטיסטית (קיבלנו כי $p \text{ value} < 0.05$) ולכן לאחר מכן הרצנו Post Hoc test על מנת לבדוק מובהקות בין האלגוריתמים. קיבלנו כי ישנה מובהקות סטטיסטית בין מודל ה- baseline לבין אלגוריתם ה- noisy student וגם כי יש מובהקות סטטיסטית בין מודל ה- baseline לאלגוריתם המשופר, כלומר התוצאות אינן רנדומליות. קיבלנו שאין מובהקות סטטיסטית בין אלגוריתם ה- noisy student לאלגוריתם המשופר, כלומר האלגוריתם המשופר אינו בהכרח משפר את תוצאות האלגוריתם מהמאמר כפי שקיבלנו ברוב ה- datasets. כאשר הרצנו עבור מקרה פרטי ללא ההנחה שביצענו כי הפרמטרים הטובים ביותר עבור אלגוריתם ה- noisy student הינם גם הפרמטרים הטובים ביותר עבור האלגוריתם המשופר קיבלנו כי האלגוריתם אכן משפר בכ- 2% אך לא היה לנו מספיק כוח עיבוד וזמן להריץ מחדש את כל התהליך של 10-fold cross validation עבור כל ה- datasets ולבצע מחדש Friedman test ולכן לא נוכל לדעת האם יש מובהקות סטטיסטית בין האלגוריתמים ללא ההנחה.