|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| \\dsp-disk.eng.tau.ac.il\Documents\eeproj\Forms & Important\פרויקטים\הנחיות\Guidelines\Students\ENG_LOGO-01.png | | | **\\dsp-disk.eng.tau.ac.il\Documents\eeproj\Forms & Important\פרויקטים\הנחיות\Guidelines\Students\TAU_EngineeringENG.png** | |
| למידת מערכות ברשת חכמה בשיטות NILM ב- ON line | | | |
| פרויקט מס' 19-1-1-1950  דו"ח סיכום | | | |
| מבצעים: | | | |
|  | מתן מצליח | 209365444 | |
|  | אלון סצ'י | 206784290 | |
| מנחה: | | | |
|  | ד"ר יובל בק | אוניברסיטת ת"א | |
| מקום ביצוע הפרויקט:  אוניברסיטת תל אביב | | | |

תוכן עניינים

[תקציר ודיאגרמת בלוקים 4](#_Toc45803041)

[1 הקדמה 5](#_Toc45803042)

[2 רקע תיאורטי 7](#_Toc45803043)

[3 שלבי ריצת הפרויקט 11](#_Toc45803044)

[4 מימוש 13](#_Toc45803045)

[4.1 מימוש חמרה 13](#_Toc45803046)

[4.2 מימוש תוכנה 14](#_Toc45803047)

[5 ניתוח תוצאות 32](#_Toc45803048)

[סיכום, מסקנות והצעות להמשך 37](#_Toc45803049)

[5.1 בחינת תוצאות הפרויקט מול המטרות שהוגדרו מלכתחילה 37](#_Toc45803050)

[5.2 הצעות לשיפור ביצועי המערכת 37](#_Toc45803051)

[5.3 עבודה עתידית על הנושא ורעיונות לפרויקטי המשך 37](#_Toc45803052)

[5.4 קישור לעמוד הגיטהאב של הפרויקט שלנו: 38](#_Toc45803053)

[מקורות 39](#_Toc45803054)

**רשימת איורים**

[איור 1 –דיאגראמת בלוקים 4](#_Toc45199518)

[איור 2 – תכנת PAS 11](#_Toc45199519)

[איור 3 – מאקרו 11](#_Toc45199520)

[איור 4 –מבנה המערכת 13](#_Toc45199521)

[איור 5 – המונה החכם 13](#_Toc45199518)

[איור 6 – סמל האפליקציה 16](#_Toc45199519)

[איור 7 – המסך הראשי 17](#_Toc45199520)

[איור 8 – מסך בחירה בין הוספה למחיקה 18](#_Toc45199518)

[איור 9 – מסך הוספה 19](#_Toc45199519)

[איור 10 – מסך הסרה 20](#_Toc45199520)

[איור 11 – דיאלוג ההסרה 20](#_Toc45199521)

[איור 12 – מסך הצגת המידע 21](#_Toc45199518)

[איור 13 – מסך הגדרות גרף הספק 22](#_Toc45199519)

[איור 14 – מסך גרף הספק 23](#_Toc45199520)

[איור 15 – מסך הגדרות גרף מכשירים 24](#_Toc45199518)

[איור 16 – מסך גרף המכשירים 25](#_Toc45199519)

[איור 17 – מסך הגדרות גרף העלות 26](#_Toc45199520)

[איור 18 – מסך גרף העלות 27](#_Toc45199521)

[איור 19 – מסך רשימת המכשירים 28](#_Toc45199518)

[איור 20 – מסך ההגדרות 29](#_Toc45199519)

[איור 21 – מסך אודות 30](#_Toc45199520)

[איור 22 – התראה על זיהוי מכשיר חדש 31](#_Toc45199519)

[איור 23 – טבלת מאורעות אמיתיים 32](#_Toc45199520)

[איור 24 – טבלת מאורעות שהמערכת גילתה 33](#_Toc45199521)

[איור 25 – גרף ההספק כתלות בזמן שנמדד לפי המונה החכם 34](#_Toc45199518)

[איור 26 – גרף ההספק שהמערכת מזהה כתלות בזמן 35](#_Toc45199519)

[איור 27 – גרף המחיר כתלות בזמן של קבוצת המכשירים מאיור 26 36](#_Toc45199520)

תקציר ודיאגרמת בלוקים

בעולם כיום, המידע הנגיש לצרכן על רשת החשמל מוגבל למידע על כמות החשמל אשר הוא משתמש בה בבית. לצרכן הממוצע לא זמינה האפשרות לזהות ולנטר מידע נוסף הנוגע לצריכת החשמל בביתו. מצופה כי מידע נוסף על רשת החשמל יאפשר לבדוק תקלות ברשת החשמל הביתית ובמוצרים של הצרכן בצורה נוחה יותר, יעודד את הצרכן לחסוך בחשמל ולנהוג בצורה יעילה וידידותית יותר לסביבה, ועוד.

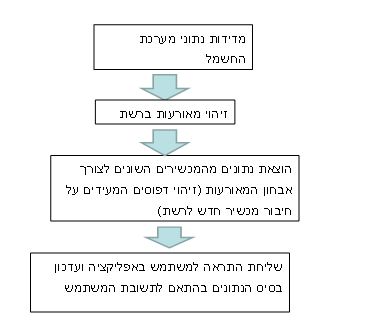
מחקרים מראים כי צרכני חשמל מורידים את צריכת החשמל החודשית שלהם בצורה משמעותית כאשר מסופק להם משוב תדיר על צריכת החשמל שלהם.

בפרויקט שלנו פיתחנו שני תוצרים:

1. אלגוריתם NILM (Non-Intrusive Load Monitoring) לזיהוי הרכיבים המצויים ברשת החשמל ולזיהוי מידע על המכשירים בהינתן מדידות שנמדדו על ידי מונה חכם של חברת SATEC.
2. אפליקציה לטלפון המתקשרת ביחד עם שרת המריץ את האלגוריתם (אשר שולח לה התראה במקרה של זיהוי עומסים חדשים) ומציגה שלל פלטים ומסקנות מתוצאותיו כמו: גרפים המציגים את העלות החודשית של צריכת החשמל של הצרכן, רשימת המכשירים שזוהו בביתו של הצרכן, הצגת גרפים של ההספק הכולל של הרשת ושל ההספק של כל מכשיר ברשת בנפרד.

המשתמש באפליקציה יכול להסיר ולהוסיף מכשירים למבנה הנתונים שלו.

הפרויקט מאפשר למשתמש להסתכל על רשת החשמל שלו ולבדוק את הצריכה של המכשירים בה ובהתאם להתאים את הרגליו בשביל להתנהג בצורה חסכונית וירוקה יותר.



איור 1 –דיאגראמת בלוקים

# הקדמה

* **מטרות הפרויקט**

מטרות הפרויקט הינן בניית אלגוריתם זיהוי מכשירים ואפליקציה קלה לשימוש, המאפשרים למשתמש לבחון את התנהגות הרשת החשמלית הביתית שלו.

בנוסף, הפרויקט מאפשר למשתמש לראות את פלטי האלגוריתם בכמה דרכים לבחירתו ולהגדיר רשימת מכשירים.

* **המוטיבציה**

יצירתם של מונים חכמים הובילה למצב בו ניתן לעקוב אחר צריכת עומסים ברשת החשמלית. היתרונות של ביצוע ניטור לצריכת העומסים הינם רבים. ישנן שתי דרכים לבצע את הניטור, דרך פולשנית (INTRUSIVE) ודרך שאינה פולשנית (Non-Intrusive), עליהן נרחיב בפרק הרקע התאורטי.

בפרויקט שלנו בחרנו להשתמש בגישה הלא פולשנית ולבנות מודל NILM.

בשנים האחרונות תחום ה – NILM מתפתח מאוד וכבר קיימות מספר לא מבוטל של חברות שמנסות לתת פתרונות לזיהוי מכשירים לצרכנים תעשייתיים וביתיים.

מכיוון שהגישה שנבחרה הינה גישה לא פולשנית קיים צורך בפיתוח אלגוריתמים המאפשרים לזהות את קיומם של עומסים חדשים ברשת החשמל וניטור מאפייניהם.

המוטיבציה בפרויקט שלנו היא לאפשר את יצירתם של שלושה רכיבים:

1. אלגוריתם הזיהוי.
2. אפליקציה המעניקה למשתמש ממשק קל לשימוש ונוח לבחינת תוצאות המעקב.
3. בסיס נתונים דינאמי של מכשירים.

* **הגישה לפתרון הבעיה**

האלגוריתם שלנו יחולק לשני חלקים, כאשר בחלק הראשון ינסה לזהות מאורעות של מכשירים מתוך המדידות שהוא קולט ברשת על ידי זיהוי קפיצות בהספק. בחלק השני, המערכת מנסה לקשר בין כל זיהוי של מאורע למכשיר קיים. זה יתבצע בעזרת פונקציית מדד בין מכשירים, כאשר ככל שהמדד גבוה יותר, המכשיר דומה יותר למכשיר שזוהה בעבר. אם הוא לא מוצא דמיון (כלומר המדד ביחס לכל המכשירים בעבר נמוך מסף מסוים) , הוא פונה למשתמש דרך האפליקציה ומבקש ממנו להכניס את המכשיר החדש שזוהה למאגר המכשירים שיושב בשרת (בסיס הנתונים הדינאמי).

הגישה הזאת תאפשר לנו למצוא קשרים בין מכשירים, מעבר למאפיינים חשמליים ( למשל הרגלי אדם) ובכך להצליח להבדיל בין מכשירים בצורה טובה. בנוסף, המערכת מעדכנת את בסיס הנתונים שלה ומידע לגבי כל מכשיר בonline, ולכן תשתפר ותלמד את הרשת בצורה חכמה.

בתחילת העבודה על הפרויקט ביצענו סקירת ספרות ולמדנו על הפתרונות הקיימים לבעיית ה – NILM. לאחר מכן ניסינו לפתח שיטות משלנו לפתרון הבעיה בהינתן נתונים שנאספו ממונה חכם של חברת SATEC. במקביל פיתחנו ממשק המאפשר למשתמשים שימוש קל במודל שנבנה בפרויקט.

# רקע תיאורטי

בעולם כיום יותר ויותר מונים חכמים מותקנים בבתים ובמקומות עבודה, דבר שמוביל לעניין גובר בשיטות לניטור עומסים לשם חסכון באנרגיה ומתן שירות איכותי יותר. ישנן שתי דרכים לניטור עומסים: ILM (Intrusive-Load-Monitoring) ו – NILM (Non-Intrusive-Load-Monitoring).

ILM פירושו שימוש במתקן ניטור לכל עומס, שיטה שהינה יקרה וקשה לתחזוק. הדבר הוביל להצגתה של שיטת ה – NILM. NILM פירושו ניטור עומסים המופעלים בו זמנית ולאחר מכן זיהוי המכשירים הדולקים. שיטה זו זולה יותר וקלה יותר לתחזוק [1].

הרעיון המקורי של NILM הוצג על ידי George W. Hart (להלן מקור [3]). Hart טען שכל מכשיר חשמלי מייצר חתימת אנרגיה שניתן להסיק ממדידות כדוגמת: מתח, זרם, והספק אקטיבי וריאקטיבי [1].

NILM מורכב מהשלבים הבאים:

* איסוף נתונים – השגת המדידות מהמונה החכם. ניתן לבצע זאת בקצב דגימה נמוך או גבוה. קצב דגימה גבוה הינו מדויק יותר, אולם קצב דגימה נמוך זול יותר וכיום נעשה בו שימוש באופן תדיר יותר.
* זיהוי אירוע והוצאת מאפיינים – השגת נתונים נוספים מהמדידות האקטיביות והריאקטיביות שהמונה החכם אסף. ניתן לחלק את הנתונים שנאספו למאפייני מצב יציב ולמאפייני מצב חולף.
* למידה והסקה – בשלב זה משתמשים באלגוריתם להוצאת מידע מהמאפיינים שהוצאו לשם זיהוי המכשירים הפועלים ובודקים האם זוהה מכשיר חדש.
* סיווג העומסים – חלוקת הצריכה הכוללת בין המכשירים והעומסים שזוהו בכדי לזהות אילו מכשירים פועלים כרגע.

בשלב זיהוי האירועים והוצאת המאפיינים ישנם כמה מאפיינים שאפשר להוציא. למשל, מאפייני מצב יציב הינם ההספק האקטיבי (הממשי) וההספק הריאקטיבי שמאפיינים כל עומס, וניתן לזהות אותם בהתבסס על דגימות בתדר נמוך של הנתונים. מאפייני מצב חולף הם מאפיינים שזוהו בהתבסס על פעולות מצב חולף של כל עומס. אולם, עבור שיטה זו נדרש דוגם תדר גבוה (כמה מגה הרצים). בנוסף, ישנם מאפיינים אחרים התלויים בשימוש אנושי יותר מאשר במאפייני המכשיר. למשל: זמן השימוש, התפלגות השימוש לאורך היום, וקורלציה בין שימושים במכשירים שונים.

בשלב הלמידה וההסקה ניתן להשתמש בכמה שיטות ואלגוריתמים. אפשר לחלק אותם לשתי קטגוריות: גישות אופטימיזציה (גישות שמתייחסות לבעיית ה – NILM כבעיית אופטימיזציה) וגישות זיהוי דפוסים (שכוללות בתוכן גישות מרקוביות וגישות של למידה חישובית מבוקרת, לא מבוקרת, ולמידה חיזוקית).

במהלך סקירת הספרות שלנו על NILM ראינו כמה גישות שונות לפתרון הבעיה אשר נדון בהן כעת. לצורכי פשטות נדון רק בגישות הרלוונטיות ביותר לטיפול בניטור עומסים.

הפתרונות שנדון בהם הם: גישת אופטימיזציה, שימוש באוטואנקודרים, שימוש במודלים מרקוביים חבויים (HMM), שימוש ברשתות קונבולוציה עמוקות (CNN), שימוש ב – CLUSTERING, שימוש בעצי החלטות, שימוש ב – SVM, שימוש בגישת אופטימיזציה לפי בעיית תרמיל הגב ,ושימוש בזכרות ארוך טווח (LSTM).

הפתרון הראשון שנדון בו הוא שימוש בגישת האופטימיזציה, אשר מתחילה בניטור מצב ה – ON/OFF של כל מכשיר ברשת החשמלית. לאחר מכן עבור כל קלט שמתקבל בודקים מה הקומבינציה של עומסים ממסד הנתונים שימזער את המרחק מהמצב שבו אנו נמצאים. שיטה זו היא השיטה הנאיבית ביותר ולא מספקת תוצאות טובות כמו שיטות אחרות (היא אינה טובה להתמודדות עם עומסים חדשים).

המודל השני שבחנו היה השימוש באוטואנקודרים, שכוללים הסתכלות על שלב הלמידה וההסקה של NILM כבעיית ניקוי רעשים. אוטואנקודר היא רשת נוירונים שמשמשת ללמידת מאפיינים באופן לא מבוקר. משתמשים בו בעיקר בניקוי רעשים מתמונות. במקורות [1] ו –[2] ראינו מחקרים שעשו שימוש בפתרון זה והגיעו ל – 55% הצלחה בזיהוי עומסים.

לאחר מכן קראנו על מודלים מרקוביים חבויים (HMM). מודלים מרקוביים חבויים הם מודלים מבוססי מכונות מצבים מרקוביות אשר בהם מצב המודל נסתר ורואים רק את פלט המודל. בהקשר של NILM המצב החבוי הינו מצב העומסים ברשת החשמל והפלט הינו הצריכה של כל עומס [1].

פתרון נוסף שחקרנו הוא השימוש ב – LSTM, שהם כלים שמשתמשים בהם בעיקר בעיבוד שפה טבעית (NLP) לצורכי מידול שפה. ל – LSTM יש חיבורי משוב והוא יכול לעבד רצפים גדולים של מידע במקום רק מידע קטן ונקודתי. נוירון LSTM מורכב מ – Gates שמשמשים לצורך בקרה על זרימת מידע [1].

במקור [4] ראינו את השימוש ב – advanced filter pipeline ובמודל עומסים גאוסיאני עם יישום הסתברותי לבעיית תרמיל הגב. בעיית תרמיל הגב הינה בעיית אופטימיזציה למציאת אוסף של פריטים (לכל אחד משקל וערך משלו) כך שנקבל את סכום הערכים המקסימלי בתנאי שלא עברנו גבול מסוים של משקל פריטים.

במקור [3] Hart הציע שימוש באלגוריתמי clustering לביצוע הקלסיפיקציה. Clustering מוגדר כחלקות אובייקטים מתוך אוסף לקבוצות בהתבסס על מאפייניהם ונקודות הדמיון בין האובייקטים. ניתן להשתמש בהרבה אלגוריתמי clustering בחלק הקלסיפיקציה של NILM כמו: BIRCH,K means, ועוד. במקור [3] Hart מציע אלגוריתם שמורכב מפיצול ואיחוד לפי בדיקות סטטיסטיות שמשתמשות בפרמטרים של הבעיה.

במקור [5] ראינו את פתרון השימוש בעצי החלטות עבור הסיווג. העצים נבנים בהתבסס על האלגוריתם הגרידי של Hunt (אלגוריתם המבוסס על הערכת נקודות הפיצול בעץ).

במקור [7] ראינו את השימוש באלגוריתם particle swarm optimization (PSO) עם support vector machines (SVMs) לצורך הקלסיפיקציה. SVM היא שיטת למידה מבוקרת שמשמשת לקלסיפיקציה ורגרסיה על כמות נתונים קטנה ו – PSO הוא אלגוריתם אופטימיזציה שמנסה לשפר את הפתרון על מדידות נתונות בצורה איטרטיבית.

הפתרון האחרון שראינו לבעיה הינו השימוש ברשתות קונבולוציה עמוקות (CNN) אשר קראנו עליו במקורות [1], [2], ו – [6]. רשתות קונובולוציה הם מחלקה של רשתות למידה עמוקה המשמשות בעיקר לעיבוד תמונה ולעיבוד שפה טבעית. ברשתות קונבולוציה עמוקות לוקחים את הקלט ומחלחלים אותו בשכבות השונות של הרשת אשר מורכבות מ - kernels (המשמשים כגלאי מאפיינים). בסוף הרשת ישנם שלושה נוירונים אשר מחזירים את זמן ההתחלה, זמן הסיום, וההספק הממוצע שנצרך.

לפי מקור [6] רשתות קונבולוציה עמוקות יכולות לשמש לזיהוי רצפי שינויים ספציפיים בהספק של העומסים ולסנן את אלה שלא הגיעו מעומסי המטרה.

לפי מקור [2] השימוש בלמידה עמוקה (רשתות קונבולוציה עמוקות ו – LSTM) הניב תוצאות טובות יותר מהמודלים המרקוביים ומגישת האופטימיזציה. מבין שתי האפשרויות פתרון מבוסס CNN החזיר את התוצאות הטובות ביותר.

**האלגוריתם שלנו**

האלגוריתם שלנו , לעומת האלגוריתמים האלו, יגדיר פונקציית מדד שלפיה יזהה את המכשירים. ההבדל העיקרי היא שהגישה שלנו היא unsupervised, כלומר אין צורך במאגר מידע קודם ולכן הגישה מאוד פרקטית, כיוון שלא דורשת זמן למידה מראש על מאגר מידע גדול של רשתות, ובפרט מאגר מידע כזה לא יכול להכיל את כל המכשירים הקיימים. נקודה נוספת שכדאי לציין היא שקשה ויקר להשיג מאגר גדול כל כך של רשתות, ובפרויקט שלנו אנחנו מנסים לתת פתרון לבעיה (חלק גדול מהפרויקט זה גם הצגה נוחה למשתמש) גם אם במחיר של חוסר דיוק מסוים.

הגישה שלנו של מדד התאמה בין מאורע למכשיר מזכירה מאוד גישה של maximum likelihood, שהיא שיטה נפוצה בסטטיסטיקה שבה מנסים להתאים מודל (או במקרה שלנו מאורע) למכשיר לפי פונקציית הסתברות (או מדד) שייתן את הציון הטוב ביותר, רק שבמקרה שלנו ציון המדד לא מייצג הסתברות.

נקודה חשובה שניתן לשים לב אליה, היא שאנחנו לוקחים פרמטרים שונים שלא מובטח שיש ביניהם קשר, ומאפיינים מכשיר לפיהם. זאת גישה שמשתמשים בה בעיקר באלגוריתמי clustering או עצי החלטה, אבל הייתרון שלנו בבניית מדד, היא שאנחנו יכולים להגדיר קשרים פנימיים בתוך המדד (למשל אם אנחנו יודעים שמכשיר תמיד מחובר לאותה נקודה, כלומר הפאזה אליה מתחבר לא תשתנה, אז אפשר להגדיר את המדד לערך מסויים, דבר שלא ניתן לעשות בclustering).

**סוגי מדידות**

ראשית כפי שציינו, המונה החכם מחובר בנקודה בודדת ברשת ולא לכל מכשיר, ולכן דוגם בעצם סכום מאפיינים של כל הרשת. המונה שלנו דוגם בקצב של דגימה כל מספר שניות, כאשר הדיוק שלו הוא עד 500kw (מציג באלפי kw), ולכן השתמשנו בדגימות הזרם שהיו עד ל0.5A (מציג באמפרים שלמים), והמרנו להספק שאמור להתקבל מהן ולכן קיבלנו דיוק טוב יותר של בערך 100kw.

מכשירים חשמליים צורכים הספק אקטיבי וריאקטיבי, כאשר מונים חכמים בדרך כלל יודעים לדגום את שניהם. באלגוריתם שלנו השתמשנו בהספק האקטיבי כאשר הגענו אליו דרך הזרם, ולעומתו לא השתמשנו בהספק הריאקטיבי, כיוון שראינו שלא נתן ערכים שונים למכשירים שונים, ולכן לא יוכל לעזור לנו בזיהוי המכשירים.

מאפיין חשמלי נוסף של רשת היא זרם ה THD(total harmonic distortion), שהיא בעצם מדידת הTHD לזרם במערכת, שהיא בעצם נוסחה שמודדת כמה האות שלנו מעוות ביחס לגל הרמוני. שמנו לב שמאפיין זה מצליח להבדיל בין מכשירים (למשל קומקום שהוא לינארי לעומת מיקרו שהוא לא לינארי), ולכן השתמשנו בו.

הפרמטר מחושב בצורה הבאה:

כאשר הן ההרמוניות המרכיבות את האות שלנו, כשהתדר של ההרמוניות עולה ככל שעולים באינדקס של ההרמוניה.

# שלבי ריצת הפרויקט

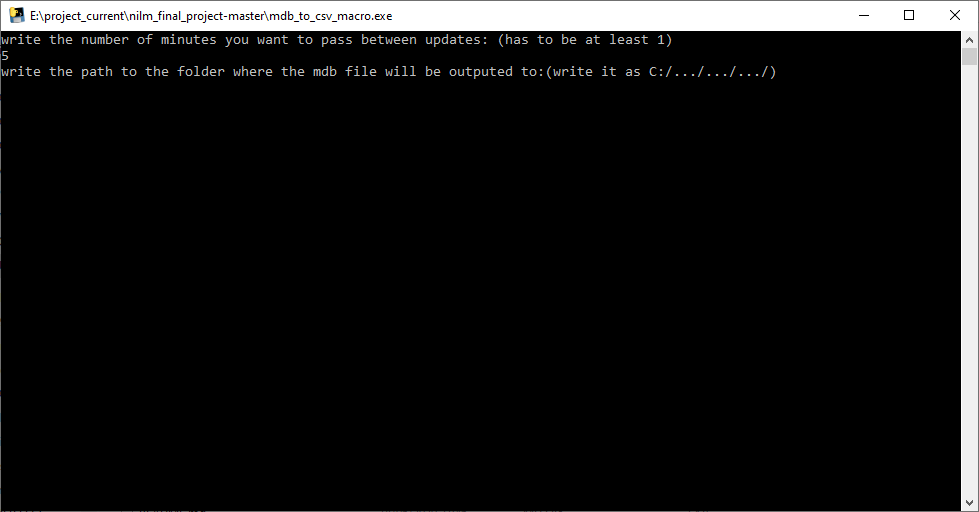
בפרק זה נתאר את שלבי ריצת הפרויקט העיקריים.

שלב 1

קבלת נתוני מדידה ממכשיר המונה החכם של חברת SATEC על ידי תכנת PAS והרצת MACRO שבנינו להמרתם לקבצי CSV.



איור 2 –תכנת PAS



איור 3 –מאקרו

שלב 2

הרצת אלגוריתם הזיהוי על המדידות. האלגוריתם יגדיר טנזור שבו כל סוגי המדידות מכל הפאזות שנפלטו מהמאקרו, יחשב קפיצות בהספק ולפיהן יבנה את המאורעות של מכשירים ברשת. לאחר מכן יבצע השוואה בעזרת פונקציית מדד למכשירים שכבר הוגדרו, ולפיהן יחליט באיזה מכשיר מדובר או האם מכשיר חדש שלא זוהה עדיין במערכת. בסופו של דבר, תתקבל רשימה של מאורעות ברשת.

שלב 3

במידה והאלגוריתם זיהה מכשיר חדש שחובר לרשת ולא זוהה במדידות קודמות, השרת ישלח לטלפון המשתמש התראה שזוהה מכשיר חדש ויש להוסיף אותו.

המשתמש יכנס לאפליקציה/ילחץ על ההתראה ויועבר למסך ההוספה, בו בוא יוסיף את המכשיר למאגר הנתונים של המכשירים שהאלגוריתם זיהה ברשת החשמלית בביתו.

שלב 4

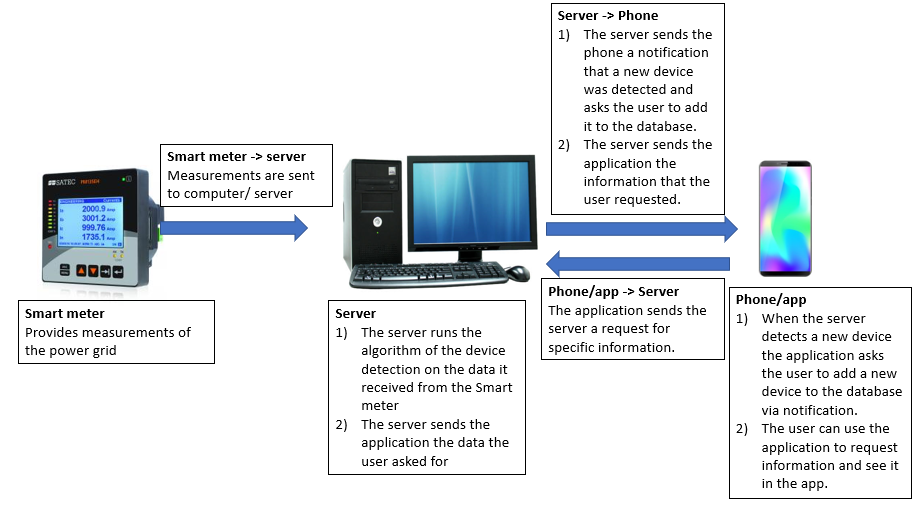
הרשת תריץ את האלגוריתם שוב בכדי לוודא שאין עוד מכשירים שנוספו.

שלב 5 (נגיש תמיד ורץ בנפרד)

המשתמש באפליקציה יוכל למחוק מכשירים ממאגר הנתונים הדינאמי ולבקש מהשרת לשלוח לו נתונים על המכשירים השונים שזוהו ועל השימוש שלו בהם בצורה של גרפים ורשימת מכשירים.

# מימוש

בפרק זה נתאר את המימוש והשיקולים לבחירתו.



איור 4 – מבנה המערכת

## מימוש חמרה

לצורך ביצוע הפרויקט שלנו היינו צריכים לבצע מדידות של ההספק האקטיבי והריאקטיבי של רשת החשמל ולצורך כך נעזרנו במונה חכם תלת פאזי רב תכליתי מסוג PM135 של חברת SATEC.



איור 5 – המונה החכם

למונה יש שתי מערכות דגימה דיגיטליות שעובדות במקביל אחת לתדר גבוה ואחת לתדר נמוך (אשר בה השתמשנו).

דוגם התדר הגבוה משיג תוצאות מדידה בדיוק גבוה.

דוגם התדר הנמוך מודד את ההספק הכולל ביחידות של KW וביחידות של KVAR ובנוסף גם את הזרם I ואת I THD (total harmonic distortion) אשר בהם השתמשנו לזיהוי המכשירים היות וההספק היה פחות מדויק.

המונה מתחבר לתוכנה בשם PAS (POWER ANALYSIS SOFTEARE) אשר מוציאה את תוצאותיו בפורמט MDB ועליה נפרט בהמשך.

את התוצאות שקיבלנו בפורמט MDB הכנסנו למאקרו שבנינו שהמיר אותן לפורמט CSV.

## מימוש תוכנה

ראשית נתאר את רכיבי התכנה שהשתמשנו בהם בבניית מימוש התכנה של הפרויקט שלנו ולאחר מכן את המימוש עצמו.

הכלים:

1. תכנת PAS – תכנה המיוצרת על ידי חברת SATEC המתממשקת ביחד עם המונה החכם ומאפשרת להוציא את תוצאות מדידתו למסמך בפורמט MDB.
2. שפת התכנה Python – שפת תכנה שפותחה בתחילת שנות ה – 90 המאפשרת כתיבת תכנית קצרה וברורה. השפה הינה מאוד קלה לשימוש ולכן גם נפוצה ביותר. קוד שנכתב בשפה רץ על מהדר ומפרש ולכן אינו דורש קומפילציה.
3. חבילת התכנה pandas – חבילת תכנה בשפת Python המיועדת לעיבוד וניתוח נתונים.
4. חבילת התכנה numpy – חבילת תכנה בשפת Python שתומכת בביצוע חישובים מתמטיים על מבנים רב ממדיים ביעילות ובמהירות.
5. חבילת התכנה matplotlib - חבילת תכנה בשפת Python המשמשת לביצוע שרטוטים (במקרה שלנו של גרפים).
6. חבילת התכנה flask – חבילת תכנה בשפת Python מסוג microframework (תשתית אפליקציית WEB מינימליסטית) המאפשרת לקוד לרוץ כשרת מסוג HTML שאפשר ליצור איתו תקשורת ובמקרה שלנו לשלוח בקשות POST ו – GET.
7. שפת התכנה JAVA – שפת תכנה שפותחה בשנות ה – 90 בחברת SUN. השפה הינה שפת תכנות מונחה עצמים ונחשבת לפופולרית מאוד. היא משמשת בין היתר (ובמקרה שלנו) ליצירת אפליקציות אנדרואיד.
8. סביבת העבודה Android Studio – סביבת עבודה בשפות JAVA ו – KOTLIN שנוצרה על ידי חברת Jetbrains לצורך יצירת אפליקציות למכשירי אנדרואיד של גוגל.
9. חבילת התכנה retrofit – חבילת תכנה בשפה java המאפשרת לתקשורת של אפליקציית האנדרואיד ביחד עם שרת במקרה שלנו לשלוח בקשות post ו – get.
10. חבילת התכנה glide - חבילת תכנה בשפה java המאפשרת להוריד תמונות מאתרים ולהציגם באפליקציה (במקרה שלנו הורדנו את התמונות שקוד ה - Python יצר מהשרת).
11. חבילת התכנה alarm manager – חבילת תכנה המאפשרת לטלפון להריץ תהליכים ברקע במקרה שלנו לשלוח לשרת פעם ב – 5 דקות הודעה שגורמת לו להריץ את האלגוריתם לצורך בדיקה האם זוהו מכשירים חדשים.
12. פורמט JSON – פורמט המייצג טקסט קריא לאדם המיועד להעברת מילון (מפתחות וערכים) בין שירותים שונים.

נפרט על רכיבי הפרויקט השונים שבנינו:

1. **מאקרו להמרת קבצי MDB לקבצי CSV** – מאקרו שנכתב בשפת Python והומר לאחר מכן לקובץ EXE. המאקרו מקבל נתיב לתיקייה כקלט וזמן דגימה, ובכל זמן מחזור מייצר קבצי CSV חדשים שמהווים המרה של קבצי ה – MDB לפורמט שקוד ה - Python שמריץ את האלגוריתם והשרת יוכל לקרוא בצורה נוחה יותר. הסיבה לכך שבנינו את המאקרו היא שקבצי המדידות שתוכנת ה – PAS מייצרת הם בפורמט MDB.
2. **אלגוריתם לזיהוי עומסים חדשים על רשת החשמל** – האלגוריתם שלנו נכתב בשפת Python. האלגוריתם תחילה קורא את קובץ הCSV שהתקבל מהמאקרו, ומגדיר ממנו טנזור של כל המדידות. לאחר מכן, האלגוריתם שלנו מזהה קפיצות בהספקים של כל אחת מ3 הפאזות, כאשר הוא מקשר בין כל עלייה בהספק לירידה בהספק לפי הגודל האבסולוטי של הקפיצה (מנסה שיהיו קרובים בערכם ככל הניתן, כאשר מוגדר סף מראש לגודל הקטן ביותר שנחשב עבורנו כקפיצה, וסף לגודל הסטייה שיכול להתקבל בהתאמת הגודל האבסולוטי של עלייה לירידה ).

לאחר מכן, הגדרנו פילטר שמוריד את כל המאורעות שהם קצרים מדי, ובכך הורדנו מקרים רבים של false positives שהמערכת שלנו זיהתה (מאורעות של מספר שניות בודדות).

לאחר מכן, קיבלנו רשימת מאורעות נקייה יותר (כמעט וללא false positives או true negatives).

אחרי זה, עברנו על רשימת כל המאורעות וחיברנו אותם למכשירים שראינו בעבר לפי פונקציית מדד, שבוססה על אינטרפולציה של הקירוב בהספקים בין המכשירים, קירוב בזרם THD (אשר מראה כמה לינארי המכשיר שלנו), קירוב ביחס לשעה ביום של מאורעות עבר של המכשיר וקירוב המרחקים בין ההתפלגות של חיבור על גבי הפאזות השונות. בנוסף הגדרנו שמכשיר יכול להיות סטטי בפאזה, ולכן המדד ייתן ציון אפסי למכשיר סטטי בפאזה ביחס למאורעות שנמדדו בפאזה אחרת.

כלומר:

כאשר הוא הציון לפי הפרמטר ה, ו הוא המשקל שנתנו עבור הציון לאותו פרמטר.

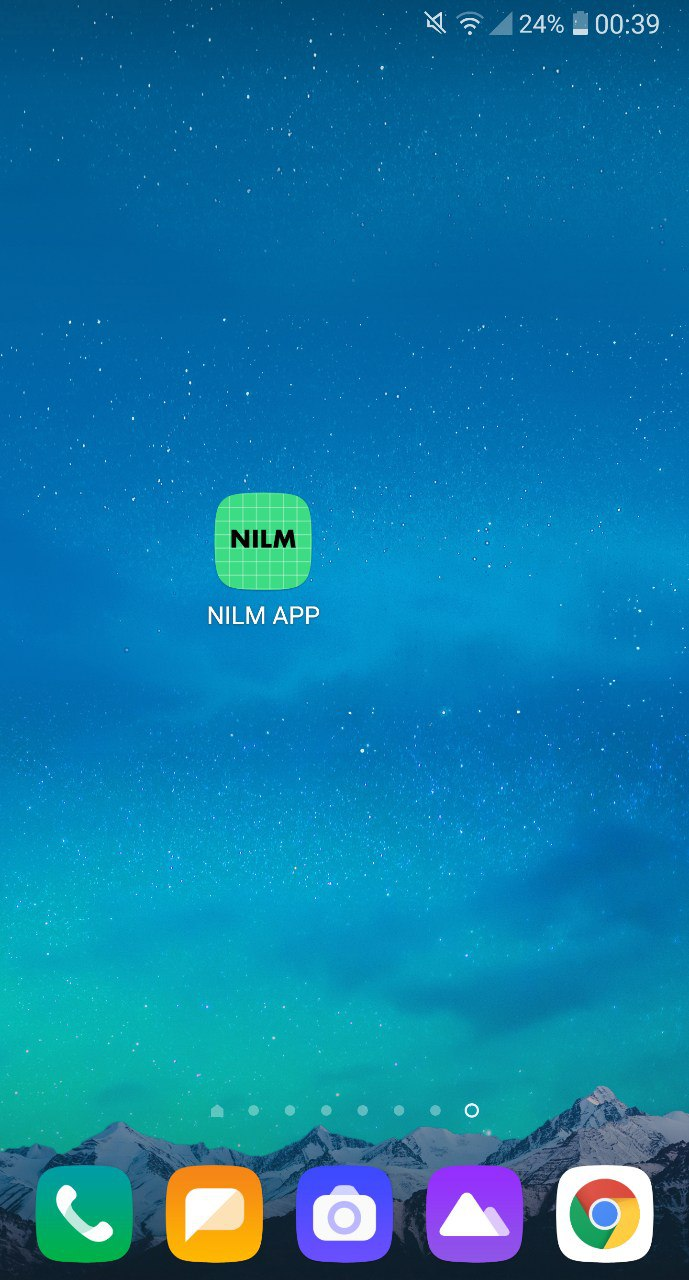
לאחר מכן, המאורעות יקושרו למכשיר בעל הציון הטוב ביותר, כאשר אם הציון הטוב ביותר עדיין נמוך מסף שהגדרנו עבור כל פאזה, נחליט כי מדובר במכשיר חדש.

1. **שרת –** השרת שנבנה בעזרת חבילת התכנה flask יושב בכתובת ה – IP של המחשב. השרת מכיל בתוכו כמה ממשקים (אתרים) אשר כל אחד מהם מקבל בקשה מהמשתמש ובהתאם מריץ את האלגוריתם או פונקציות אחרות שנותנות למשתמש מידע שהוסק כתוצאה מהמדידות ומהמכשירים שזוהו. את השרת בנינו בפורמט HTTPS בגלל מגבלות של מערכת אנדרואיד (החל מגרסאות חדשות של המערכת) המאפשרות רק לתקשורת מאובטחת לעבור מאפליקציות ללא שליחת התראה למשתמש.

ממשקי השרת הם:

* + “https://IP:5000/app” - ממשק שלאחר קבלת בקשה מהמשתמש מריץ את האלגוריתם ומנסה לזהות האם יש מכשירים שלא זוהו ברשת. במידה וישנם כאלה הממשק ישלח לאפליקציה הודעה שזוהו מכשירים חדשים ויש להוסיף אותם למאגר המכשירים הדינאמי.
  + “https://IP:5000/add\_device\_json” - ממשק שמקבל מהמשתמש את המידע על המכשיר שיש להוסיף בפורמט JSON ובכך מנתח את המידע. אם לא זוהה מכשיר עם אותו השם במבנה הנתונים השרת מוסיף את המכשיר עם המידע שהמשתמש הכניס והמידע שזוהה על ידי האלגוריתם לתוך רשימת המכשירים. במידה וקיים מכשיר בעל שם זהה במאגר המשתמש יקבל הודעה שתגיד לו שמכשיר עם שם זה קיים ושעליו לנסות שם אחר.
  + “https://IP:5000/remove\_device\_json” - ממשק שמקבל מהמשתמש שם של מכשיר שהמשתמש רוצה להסיר מרשימת המכשירים, במידה ולא זוהה מכשיר בעל שם זה המשתמש יקבל על כך התראה אחרת המכשיר ימחק מרשימת המכשירים.
  + “https://IP:5000/nilmgraph\_power” - ממשק שמקבל מהמשתמש טווח זמנים ומשרטט גרף של ההספק בהתאם לזמנים שהמשתמש הכניס.
  + “https://IP:5000/nilmgraph\_devices” - ממשק שמקבל מהמשתמש טווח זמנים, ורשימה של מכשירים שהמשתמש בחר מתוך הרשימה הכוללת ומשרטט גרף של ההספק של המכשירים שהמשתמש בחר בהתאם לזמנים שהמשתמש הכניס.
  + “https://IP:5000/nilmgraph\_cost” - ממשק שמקבל מהאפליקציה טווח זמן ומחיר ביחידות של KWH למטבע שהוא בחר לשלם בו (או בקשה לפשוט לראות כמה KWH הוא השתמש בפרק הזמן) ומחזיר למשתמש גרף עמודות של עלות צריכת החשמל שלו בפרק הזמן הזה כאשר כל עמודה מייצגת חודש.
  + “https://IP:5000/get\_device\_list” - ממשק שמקבל מהמשתמש בקשה ושולח לו את רשימת המכשירים בפורמט JSON כדי שהאפליקציה תוכל להציג אותה.

1. **אפליקציה** – אפליקציה למכשירי אנדרואיד המתקשרת ביחד עם השרת ומאפשרת הצגת המידע שהאלגוריתם הסיק כתוצאה מנתוני השרת והזנת מידע נוסף שיסייע לאלגוריתם (הזנת מכשיר חדש לרשימה). האפליקציה מורכבת משלל activities (דבר שדומה למחלקות בשפת JAVA המשמש באפליקציות אנדרואיד לביצוע פעולות שונות ולהצגת מסכים שונים) ו – framents (סוג של subactivities).



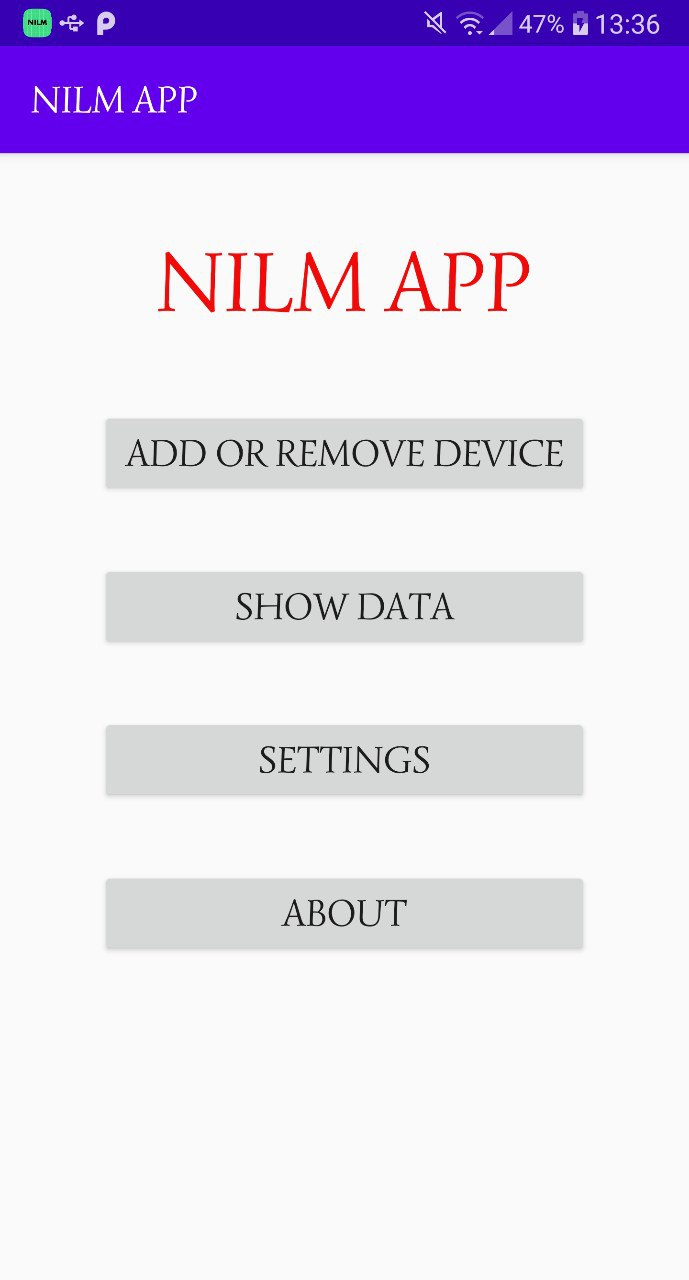
איור 6 – סמל האפליקציה

בגלל אבטחת מכשירי אנדרואיד והעובדה שהרצנו את השרת על המחשבים הביתיים שלנו ולא על דומיין שקנינו, לא היה לנו סרטיפיקט SSL ולכן בשביל לאפשר תקשורת של השרת עם האפליקציה דרסנו את בנאי התקשורת הדיפולטי של אנדרואיד בהקשר האפליקציה כך שלא יבדוק את הסרטיפיקט, אלא פשוט יקבל את התקשורת לא משנה האם לשרת יש סרטיפיקט אמין.

להצגת התמונות נעזרנו בחבילת תכנה בשם GLIDE שנותנת את היכולת להוריד תמונות מהאינטרנט ולהציג אותן.

פעילויות האפליקציה:

מסך ראשי- מסך המאפשר לבחור בין הוספת/הסרת מכשיר (ADD OR REMOVE DEVICE), הצגת המידע/הנתונים (SHOW DATA), הגדרות (SETTINGS), אודות האפליקציה (ABOUT).



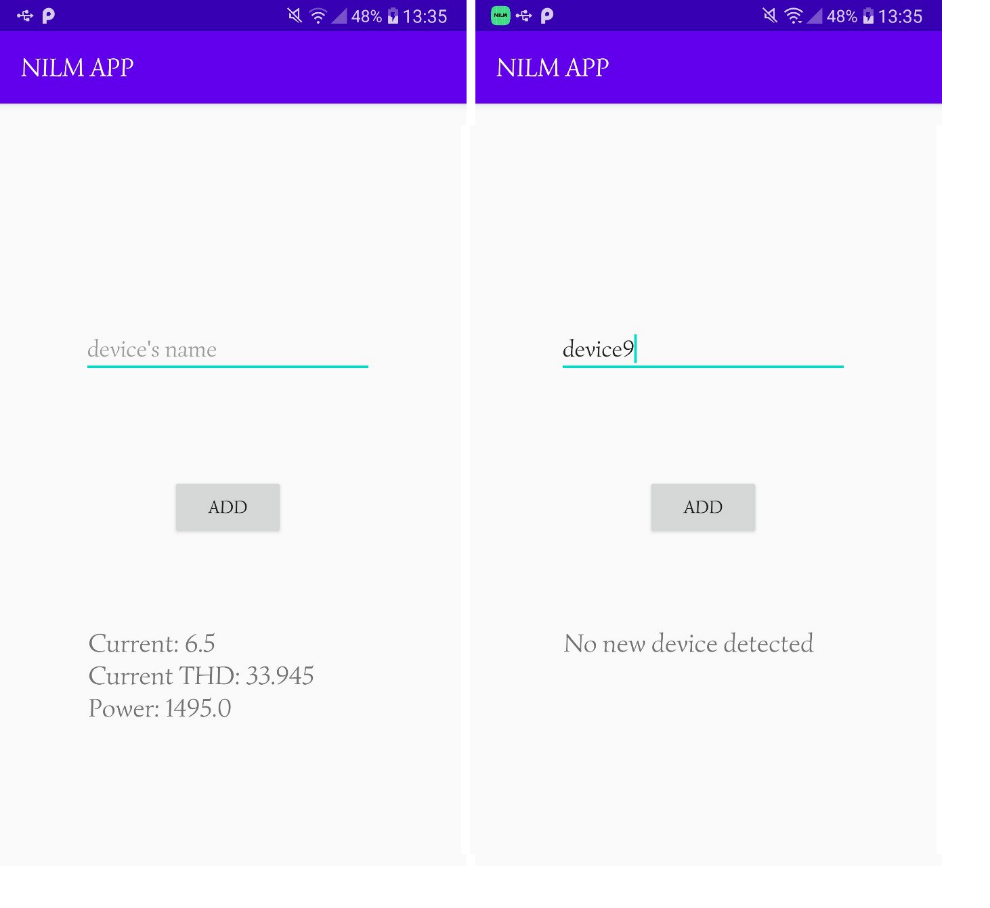
איור 7 – המסך הראשי

מסך בחירה בין הוספה למחיקה (הגענו אליו ממסך ראשי על ידי לחיצה על ADD OR REMOVE DEVICE) - מסך המאפשר לבחור בין האם אנחנו רוצים להוסיף לרשימה מכשיר (ADD A DEVICE) לבין האם אנחנו רוצים למחוק מכשיר מהרשימה (REMOVE A DEVICE).



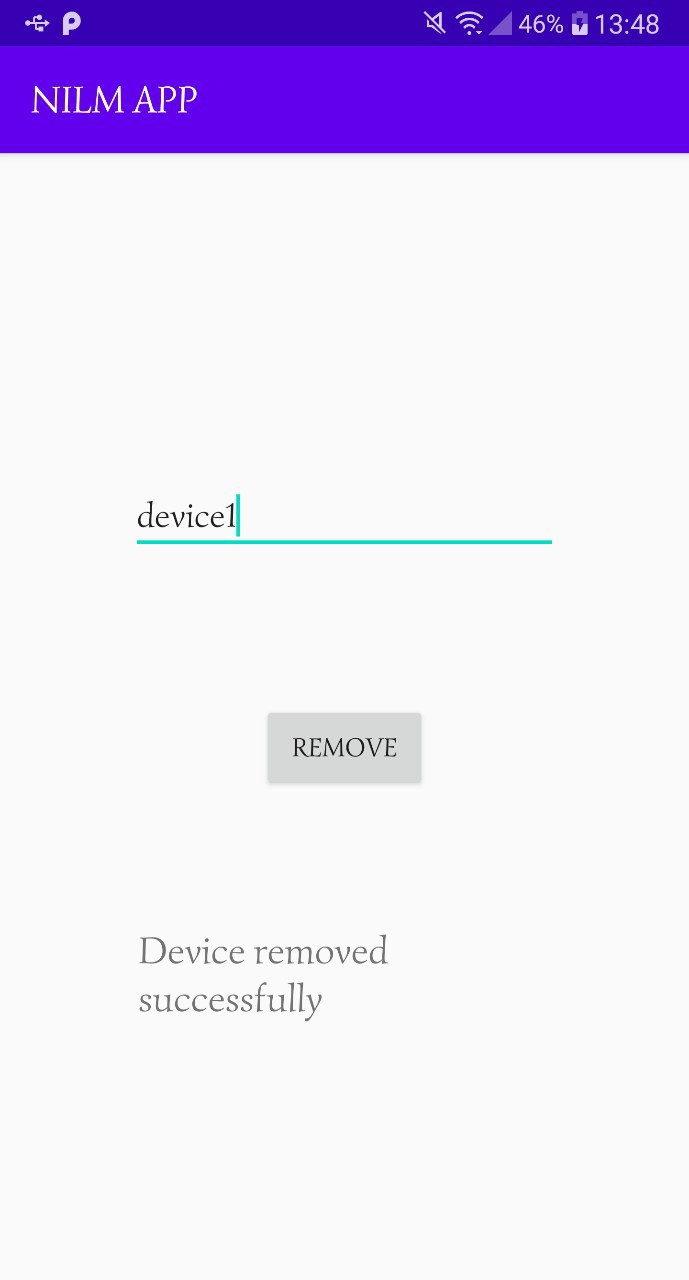
איור 8 – מסך בחירה בין הוספה למחיקה

הוספת מכשיר (הגענו אליו על ידי שתי אפשרויות לחיצה על התראה שזוהה מכשיר חדש או על ידי לחיצה על ADD A DEVICE במסך ADD OR REMOVE A DEVICE)- מסך ששולח לשרת בקשה ששואלת האם יש מכשיר חדש שזוהה ואם כן מבקש את הפרטים ומציג אותם בתחתית המסך. כאשר המשתמש רואה את פרטי המכשיר הוא מכניס שם למכשיר לשורת הטקסט ולוחץ על ADD. לאחר מכן נשלחת לשרת הודעה שאומרת לו להוסיף מכשיר עם השם ששלחנו והפרטים שזוהו. אם מכשיר עם שם זה כבר קיים בשרת תרשם למשתמש הודעה שמכשיר עם שם זה קיים ושינסה לשים שם אחר למכשיר. אם לא זוהו מכשירים נוספים שיש להוסיף תרשם למשתמש הודעה שאומרת שלא זוהו מכשירים חדשים.

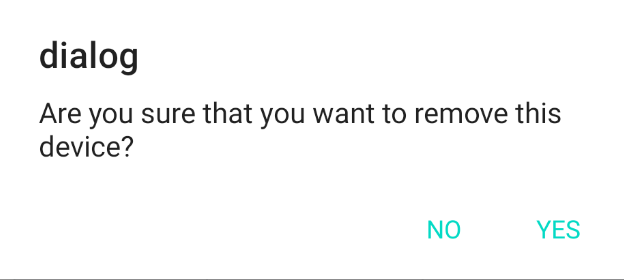


איור 9 – מסך הוספה

הסרת מכשיר (הגענו אליו על ידי לחיצה על REMOVE A DEVICE במסך ADD OR REMOVE A DEVICE)- במסך זה המשתמש מכניס לשורת הטקסט שם של מכשיר ולאחר מכן לוחץ על REMOVE ונפתח לו דיאלוג שמוודא שהוא מעוניין למחוק את המכשיר. במידה והוא לוחץ שכן תשלח הודעה על כך לשרת. אם זוהה מכשיר עם שם זה ברשימה הוא נמחק ממנה והמשתמש מקבל הודעה שהמכשיר נמחק בהצלחה. אם לא זוהה מכשיר עם שם זה ברשימת המכשירים המשתמש מקבל הודעה שלא זוהה מכשיר העונה לשם שהמשתמש הכניס.

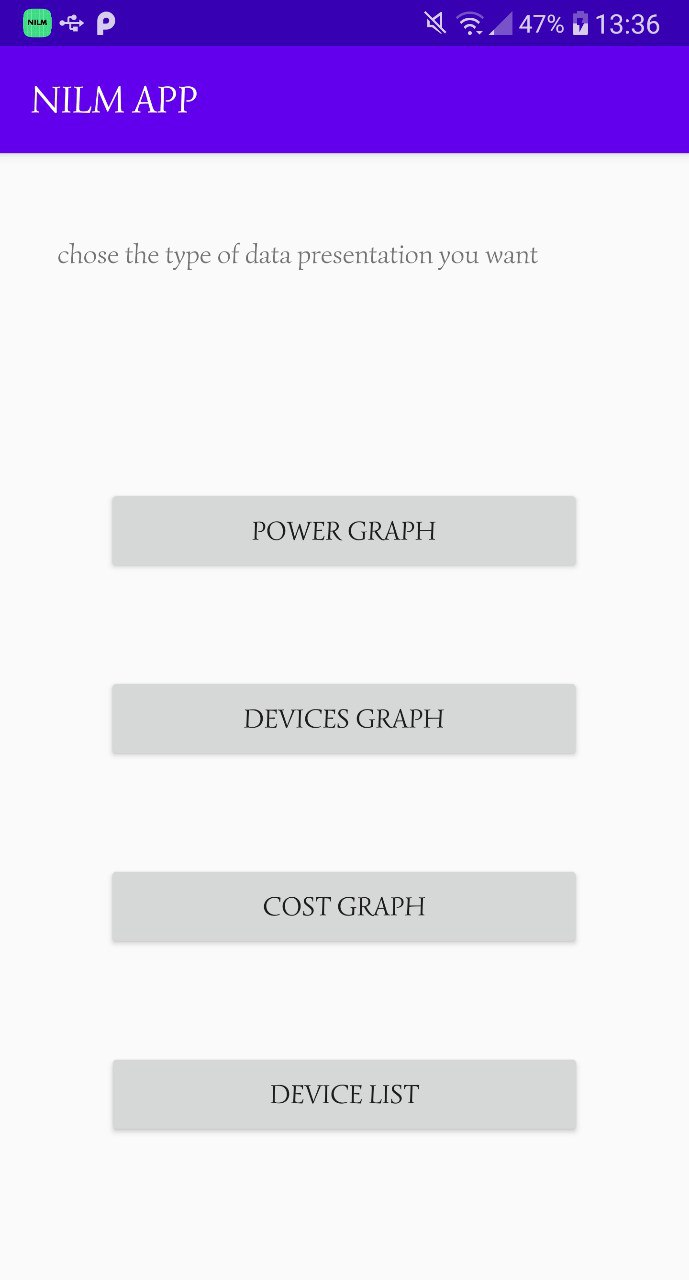


איור 10 – מסך הסרה



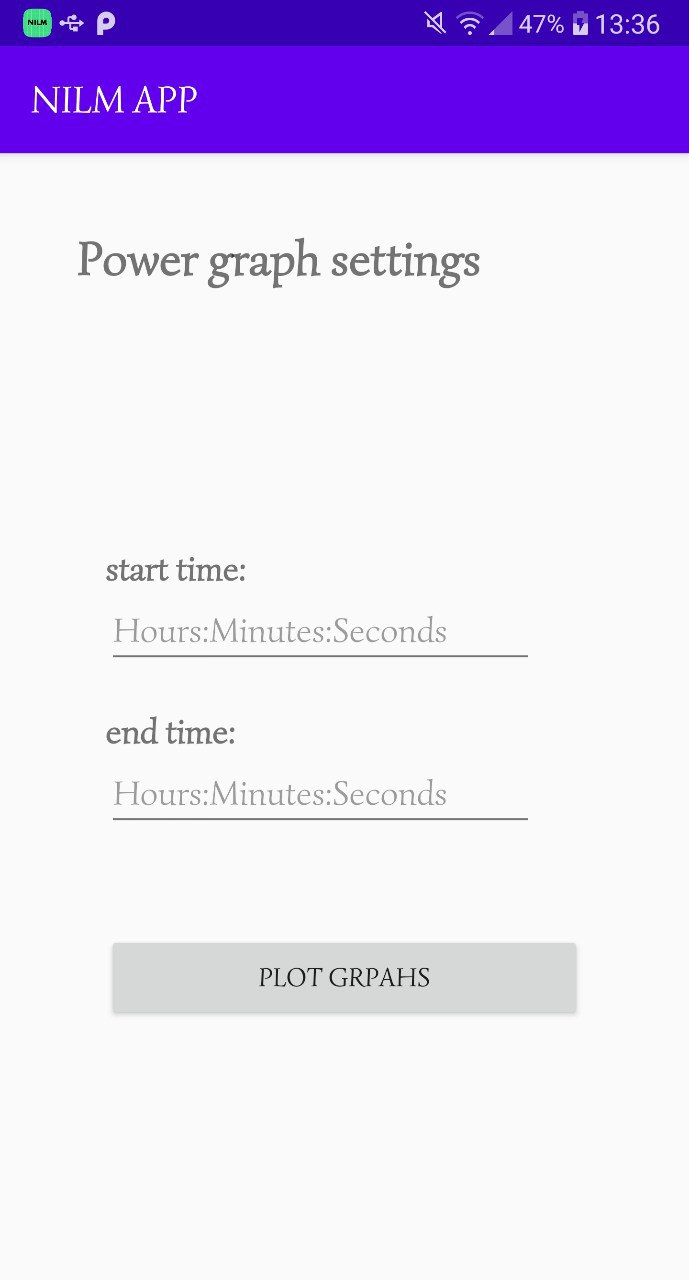
איור 11 – דיאלוג ההסרה

מסך הצגת המידע (הגענו אליו על ידי לחיצה על SHOW DATA במסך הראשי) – המסך מאפשר למשתמש לבחור איזה מידע הוא רוצה לראות מהאפשרויות – גרף הספק (POWER GRAPH), גרף הספק פר מכשיר (DEVICES GRAPH), גרף עלות/מחיר שמציג את המידע לגבי עלות הצריכה החודשית של המשתמש (COST GRAPH), והצגת רשימת המכשירים שזוהו (DEVICE LIST).



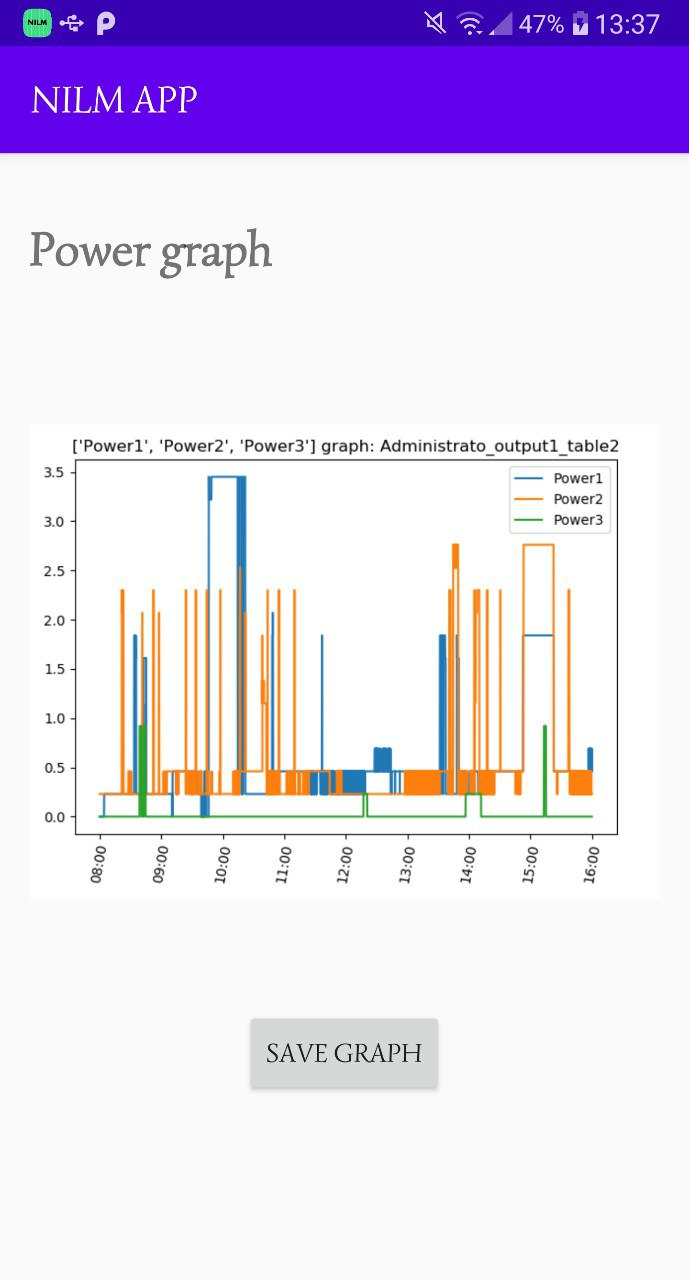
איור 12 – מסך הצגת המידע

מסך הגדרות גרף הספק (הגענו אליו על ידי לחיצה על POWER GRAPH במסך הצגת המידע) – במסך זה המשתמש מכניס לשורות הטקסט את טווח השעות שיוצג בגרף ההספק, ולאחר מכן לוחץ על PLOT GRAPHS. לאחר הלחיצה חבילת JSON שמכילה את הבקשה ואת טווח השעות תשלח לשרת שיעבד את המידע שקיבל.



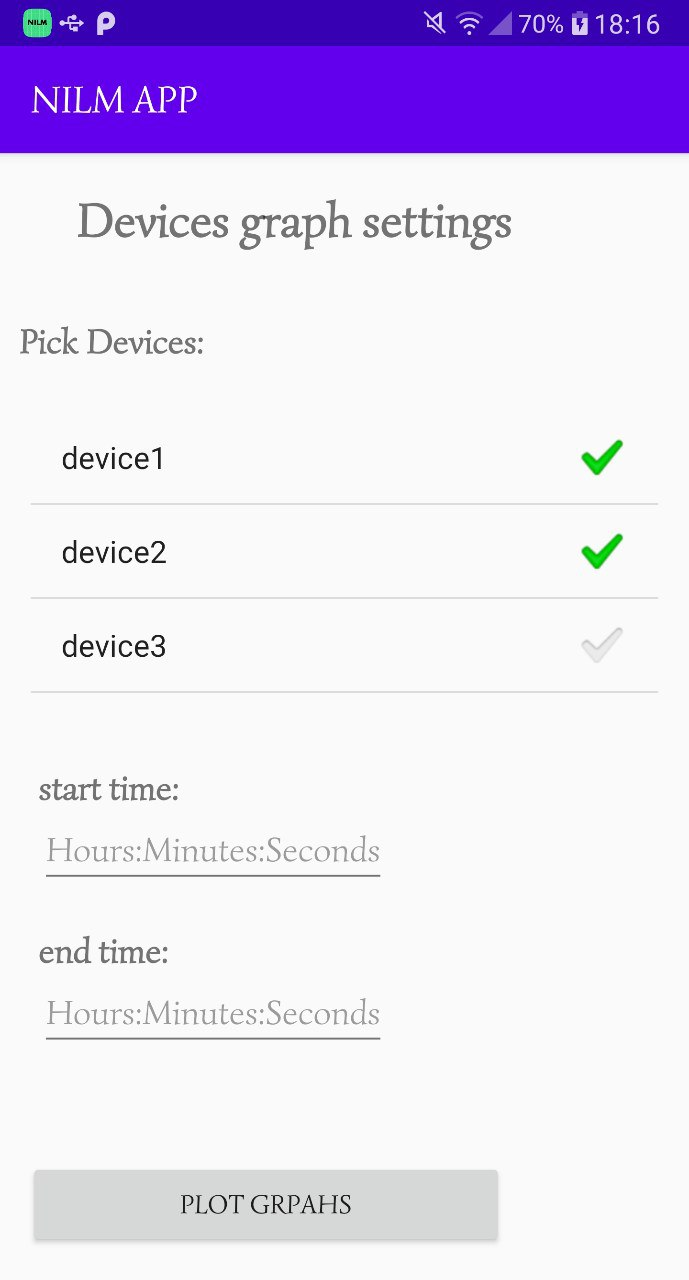
איור 13 – מסך הגדרות גרף הספק

מסך גרף ההספק (הגענו אליו על ידי לחיצה על PLOT GRAPHS במסך הגדרות גרף ההספק) – מסך שמציג את גרף ההספק בטווח השעות שהכנסנו על ידי שימוש ב – GLIDE להורדת התמונה של הגרף מהשרת. במסך ישנו כפתור SAVE GRAPH. לחיצה עליו תשמור את הגרף בתוך תיקייה בגלריית התמונות בטלפון של המשתמש.



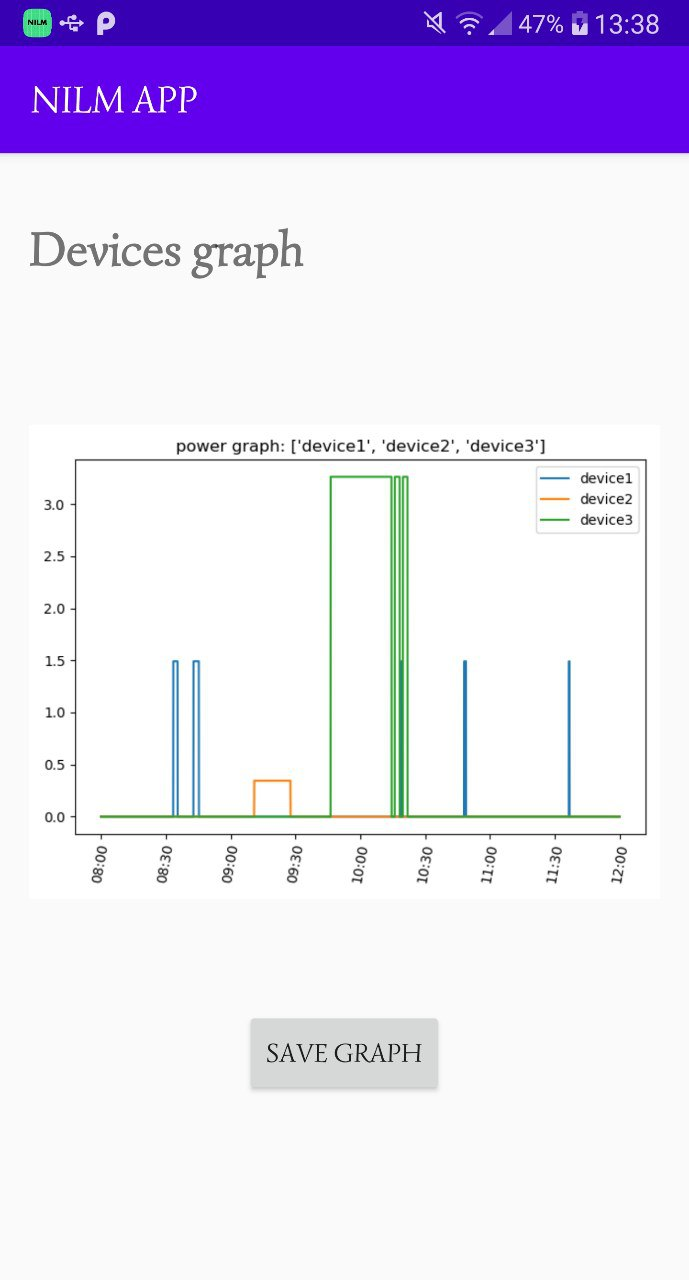
איור 14 – מסך גרף הספק

מסך הגדרות גרף מכשירים (הגענו אליו על ידי לחיצה על DEVICES GRAPH במסך הצגת המידע) – במסך זה המשתמש בוחר את המכשירים (בעזרת ה – CHECKBOXES) שהוא מעוניין לראות בגרף ההספק לפי מכשיר שהוא רוצה לקבל מהשרת ואת טווח הזמנים של הגרף. לאחר שהכניס את הנתונים הוא לוחץ על PLOT GRAPHS.



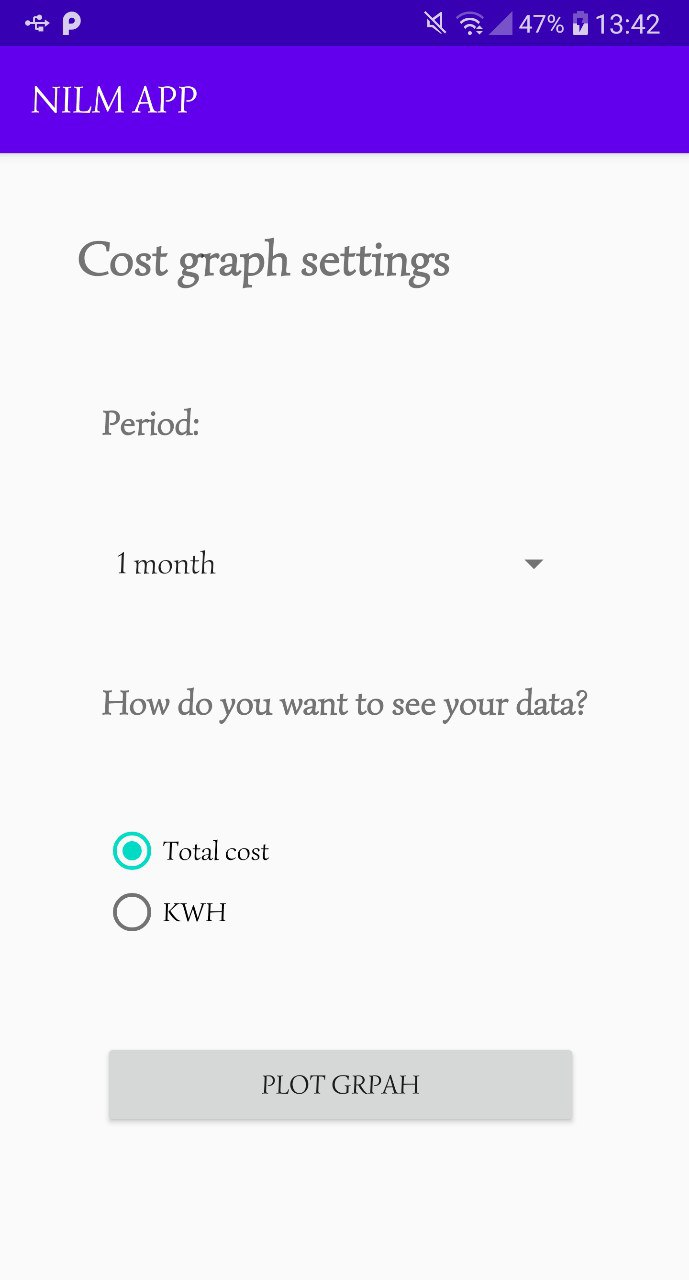
איור 15 – מסך הגדרות גרף מכשירים

מסך גרף המכשירים (הגענו אליו על ידי לחיצה על PLOT GRAPHS במסך הגדרות גרף המכשירים) – מסך שמציג את גרף ההספק בטווח השעות לפי מכשיר שהכנסנו על ידי שימוש ב – GLIDE להורדת התמונה של הגרף מהשרת. במסך ישנו כפתור SAVE GRAPH. לחיצה עליו תשמור את הגרף בתוך תיקייה בגלריית התמונות בטלפון של המשתמש.



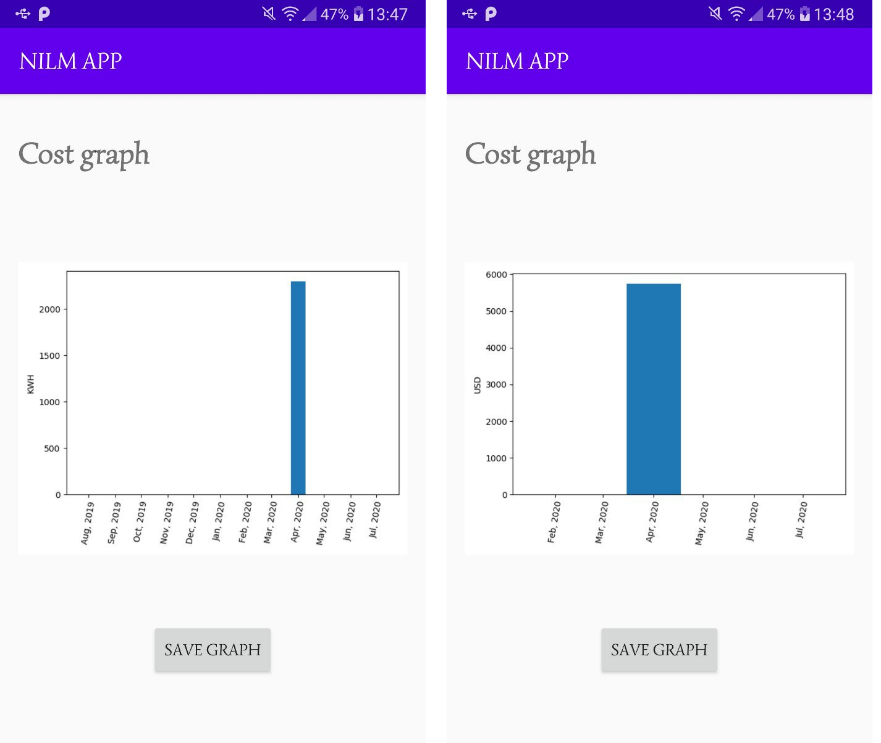
איור 16 – מסך גרף המכשירים

מסך הגדרות גרף העלות (הגענו אליו על ידי לחיצה על COST GRAPH במסך הצגת המידע) – במסך זה המשתמש בוחר את טווח החודשים ביחס לחודש הנוכחי שהוא רוצה לראות בגרף ואת היחידות (האם הוא רוצה לראות את עלות החשמל החודשית שלו או רק כמה KWH הוא מבזבז בחודש). לאחר שהכניס את הנתונים הוא לוחץ על PLOT GRAPHS. המשתמש יכול לבחור במסך זה בין גרף של החודש האחרון, החודשיים האחרונים, שלושת החודשים האחרונים, חצי השנה האחרונה, השנה האחרונה והשנתיים האחרונות.



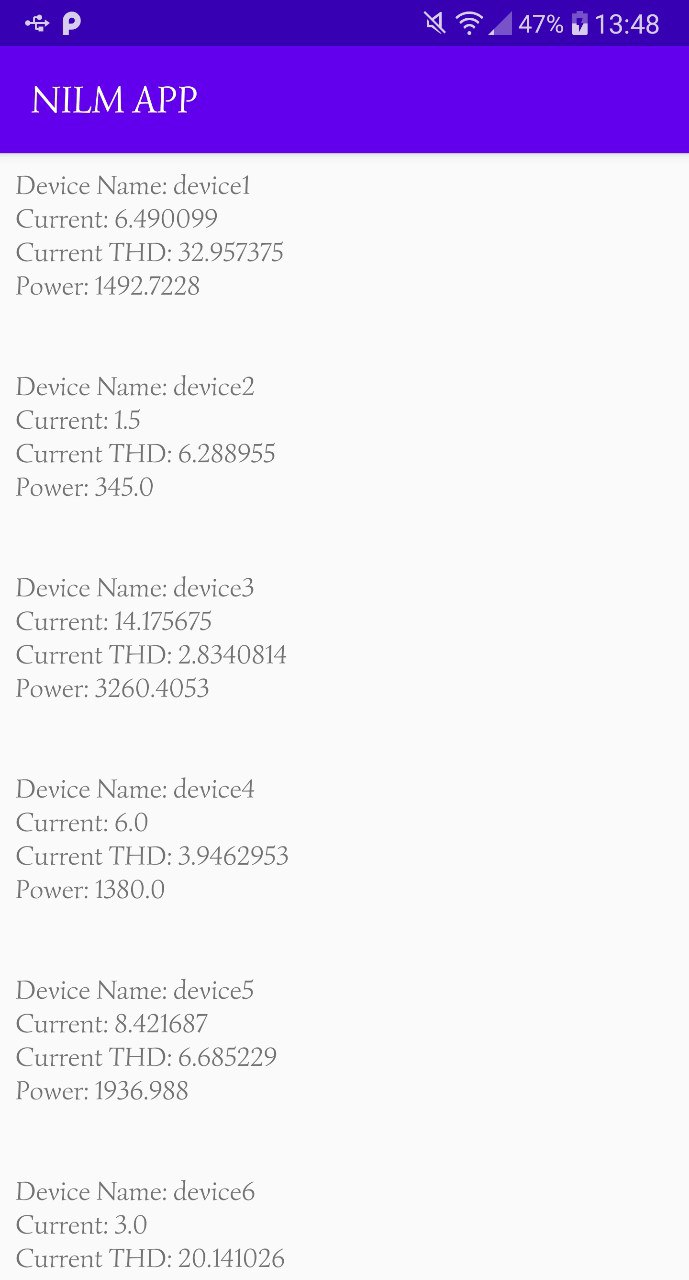
איור 17 – מסך הגדרות גרף העלות

מסך גרף העלות (הגענו אליו על ידי לחיצה על PLOT GRAPHS במסך הגדרות גרף העלות) – מסך שמציג גרף עמודות של עלות החשמל החודשית של המכשירים שהאלגוריתם זיהה לפי נתוני צריכת המשתמש מהמונה החכם. את הגרף האפליקציה קיבלה על ידי שימוש ב – GLIDE להורדת התמונה של הגרף מהשרת. במסך ישנו כפתור SAVE GRAPH. לחיצה עליו תשמור את הגרף בתוך תיקייה בגלריית התמונות בטלפון של המשתמש.



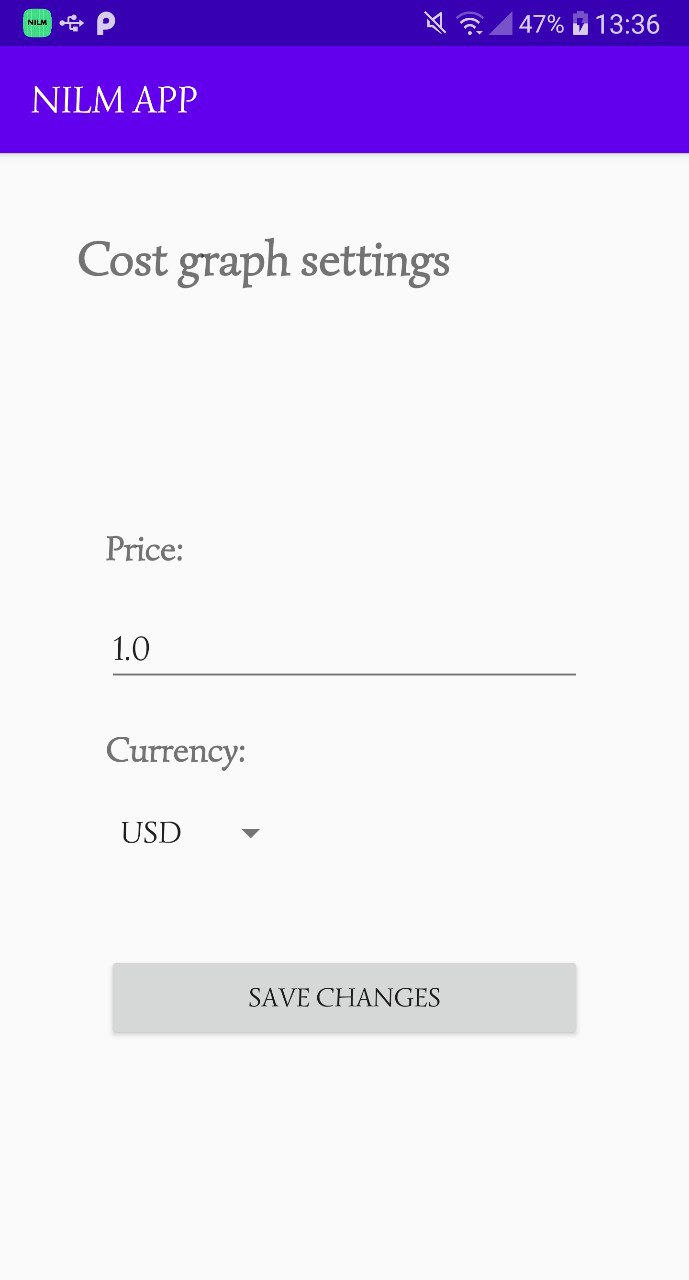
איור 17 – מסך גרף העלות

מסך רשימת המכשירים (הגענו אליו על ידי לחיצה על DEVICE LIST במסך הצגת המידע) – מסך זה מקבל מהשרת את רשימת המכשירים שזוהו על ידי האלגוריתם ונוספו על ידי המשתמש ומציג לו אותה.



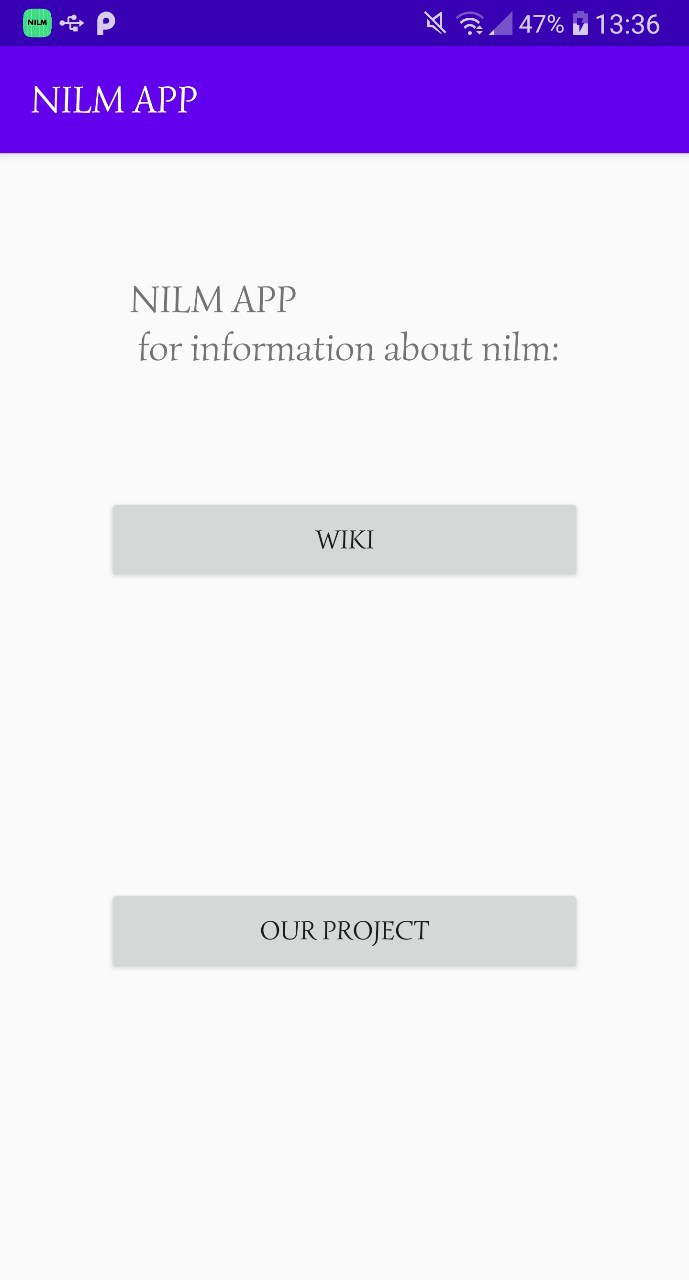
איור 19 – מסך רשימת המכשירים

מסך ההגדרות (הגענו אליו מהמסך הראשי על ידי לחיצה על SETTINGS) – במסך זה המשתמש מכניס את העלות של צריכת חשמל ביחידות של KWH למטבע כאשר הוא בוחר במטבע מתוך תפריט dropdown שמכיל את האפשרויות USD,EUR,ILS,BTC,GBP,BTC,ETH (סוגי מטבעות פופולריים).



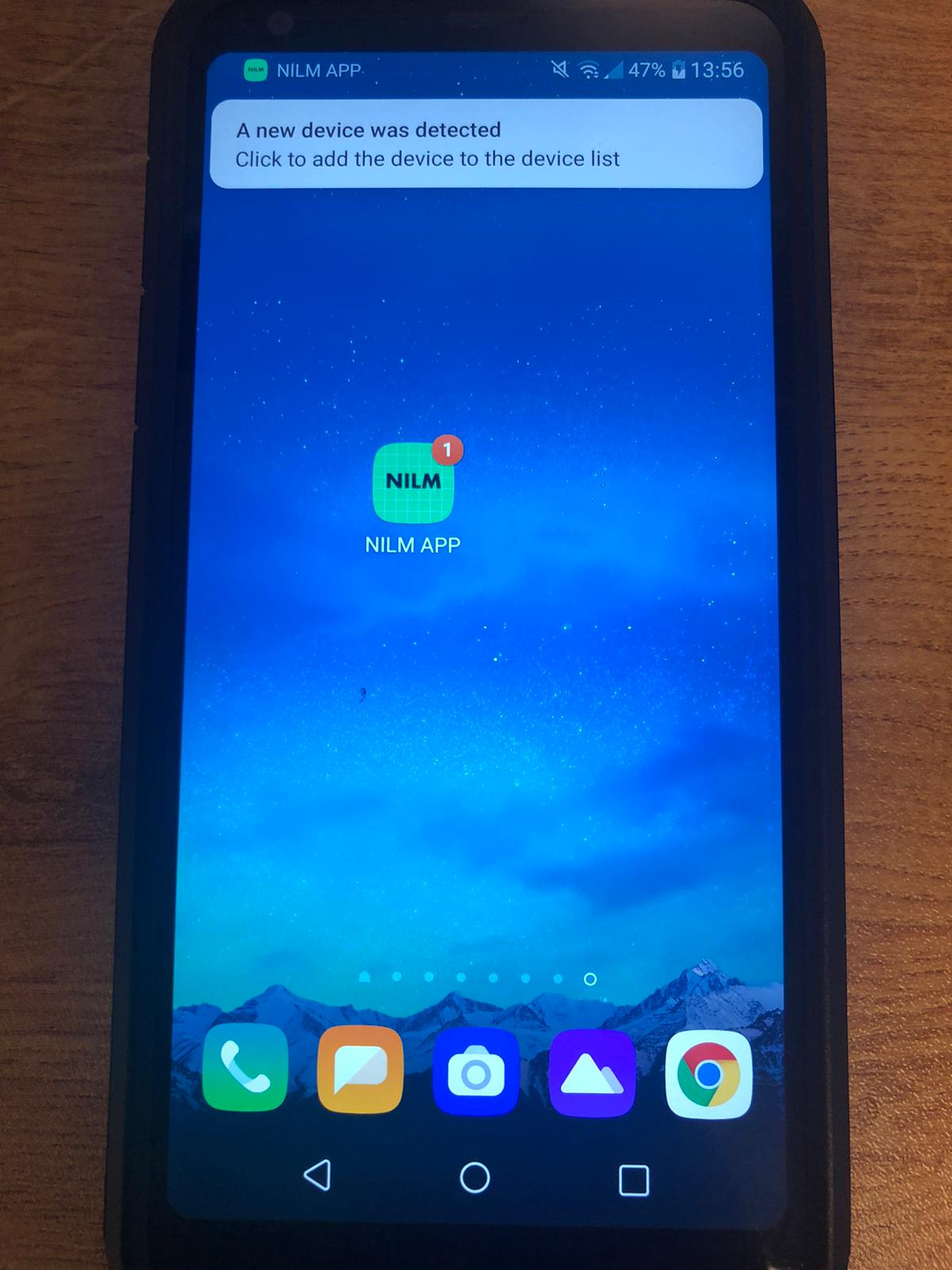
איור 20 – מסך ההגדרות

מסך מידע אודות האפליקציה (הגענו אליו על ידי לחיצה על ABOUT במסך הראשי) – במסך זה המשתמש יכול לקבל מידע על NILM ועל האפליקציה כאשר יש לו שתי אפשרויות. הראשונה מובילה אותו לדף הויקיפדיה של NILM והשנייה לדף הגיט האב של הפרויקט שלנו.



איור 21 – מסך אודות

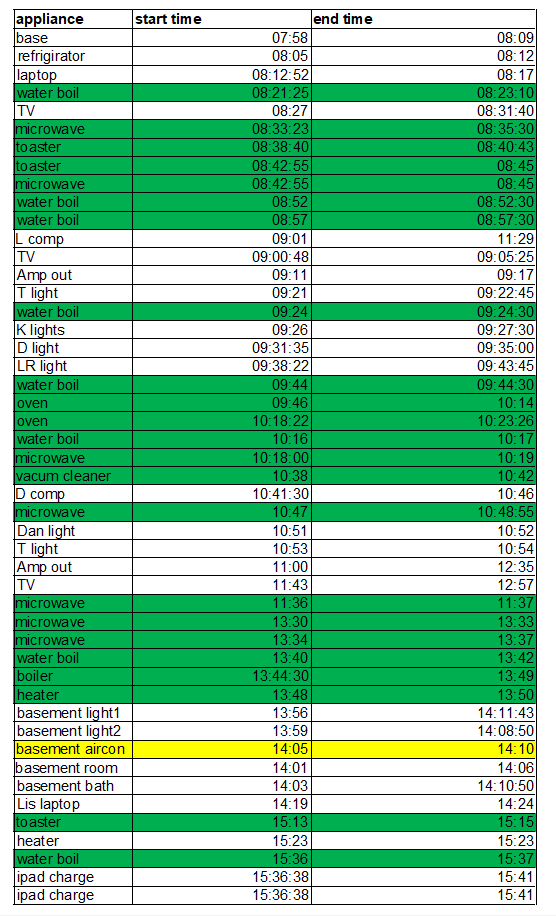
התראת זיהוי מכשיר חדש (מתקבלת כאשר האלגוריתם מזהה מכשיר חדש והשרת מודיע על כך) – בכל פרק זמן של 5 דקות האפליקציה (גם אם היא לא רצה התהליך הזה מורץ ברקע) שולחת לשרת הודעה שמבקשת ממנו להריץ את האלגוריתם. במידה והאלגוריתם זיהה מכשיר חדש הוא מודיע על כך למשתמש על ידי שליחת הודעה על זיהוי מכשיר חדש שמפעילה את ההתראה בטלפון. כאשר המשתמש לוחץ על ההתראה הוא מופנה למסך הוספת המכשיר. בכדי שלא יופיעו למשתמש התראות רבות (כך שיהיה לו יותר נוח עם האפליקציה) בכל פעם שבה נשלחת התראה היא דורסת התראה קודמת כך שבפרק זמן נתון מוצגת למשתמש לכל היותר התראה אחת.



איור 22 – התראה על זיהוי מכשיר חדש

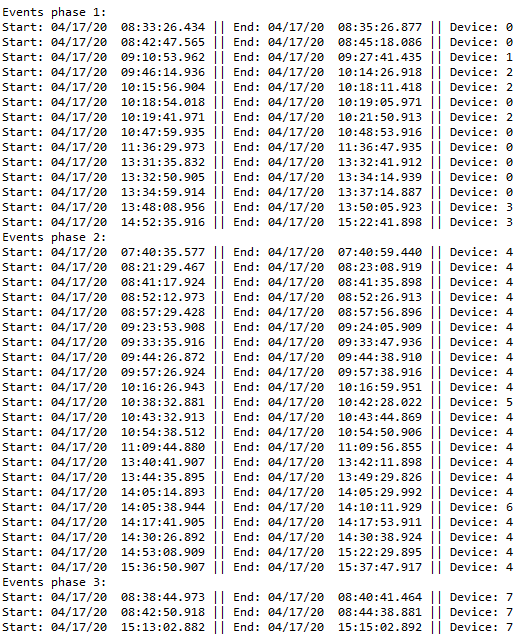
# ניתוח תוצאות

נציג את תוצאות המערכת שלנו על אוסף דגימות במשך חצי יום (גודל טווח הזמן שהמונה החכמה מאפשר להוציא כטבלה):



**איור 23 - טבלת מאורעות אמיתיים (בירוק מאורעות שהאלגוריתם זיהה בצורה נכונה, בצהוב מאורעות שגילה וקישר למכשיר שגוי, בלבן מאורעות שלא גילה)**

כעת נציג את תוצאות המערכת (זיהוי וקישור מאורעות):



**איור 24 - טבלת מאורעות שהמערכת גילתה (בכל שורה זמן התחלה, זמן סיום, ומכשיר מקושר)**

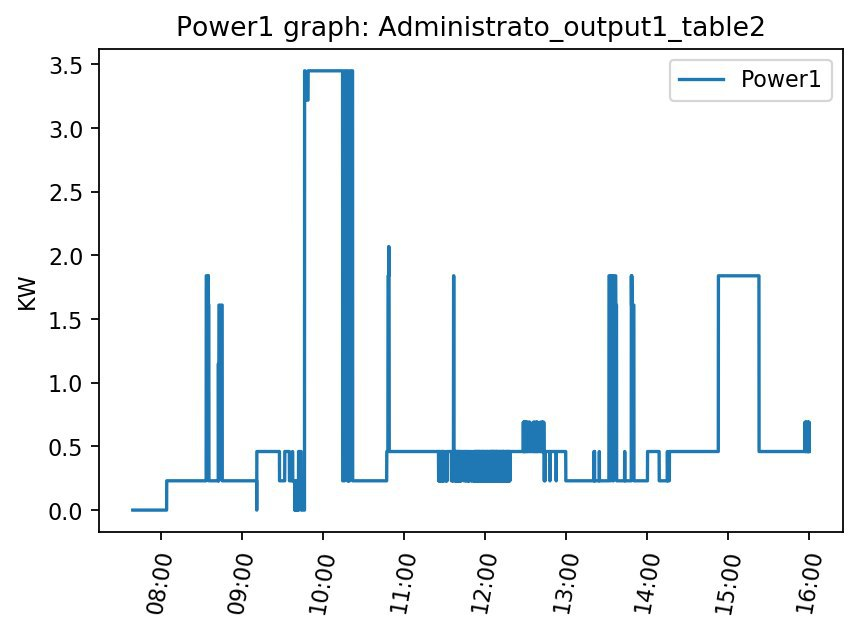
נשים לב, ראשית שהמערכת שלנו הצליחה באופן תקין לגלות ולזהות חלק גדול מהמאורעות שלנו (איור 22). כאשר, כל המאורעות שלא זיהה היו בעלי הספק נמוך מאוד, אשר האלגוריתם שלנו סינן בתהליך גילוי המאורעות (כאשר משנים את הסף המינימלי לגילוי מאורע- כך שמאורעות אלו יקלטו, מתקבלים מקרים רבים מאוד של true negatives, ולכן נשארנו עם הסף הזה).

כאשר מסתכלים על כל המאורעות אשר ההספק שלהם גדול מהסף ההתחלתי שקבענו, נשים לב שרק במקרה בודד המערכת קישרה מאורע למכשיר שגוי (השורה הצהובה באיור 22), כאשר קישר בין מזגן לבוילר. כאשר הסתכלנו בדגימות, ראינו שהמאפיינים החשמליים שלהם היו דומים מאוד, וזאת הסיבה שקישר ביניהם.

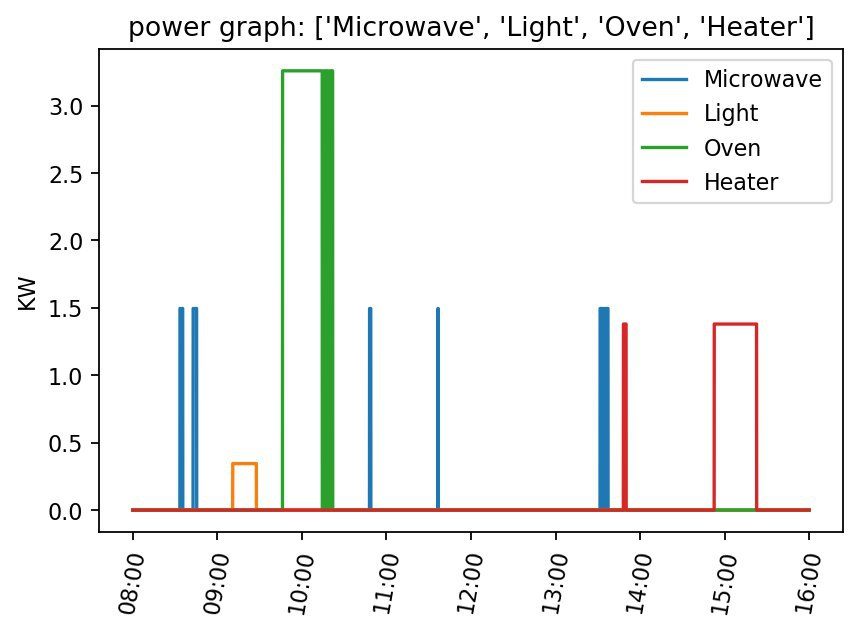
בסך הכול המערכת שלנו הצליחה להגדיר 8 מכשירים שונים על הרשת שלנו, ואנו מניחים שאם הייתה רצה במשך תקופה ארוכה על רשת, הייתה פועלת בצורה טובה יותר, כיוון שהחלק במדד שבו מבצעים הבחנה לפי זמן השימוש ביום, לא יכול לבוא לידי ביטוי באוסף הדגימות הנתון, שמכשירים בו הודלקו וכובו במיוחד כדי ליצור מאגר מידע שמכיל מספר רב של מכשירים אך לא משקף חיי יום יום של אדם.

בנוסף, האלגוריתם שלנו יכול לרוץ בזמן אמת (רץ במשך מספר שניות ספורות על מאגר מידע של עשרות אלפי דגימות), ולכן יכולנו באמצעותו לבנות ממשק משתמש נוח שבזמן אמת מחשב את הקישור של המכשירים.

נעבור כעת לגרפים ונראה שהם מציגים טוב את המאורעות. נסתכל רק על הפאזה הראשונה כדי שנוכל להציג את ההבדלים בקלות:



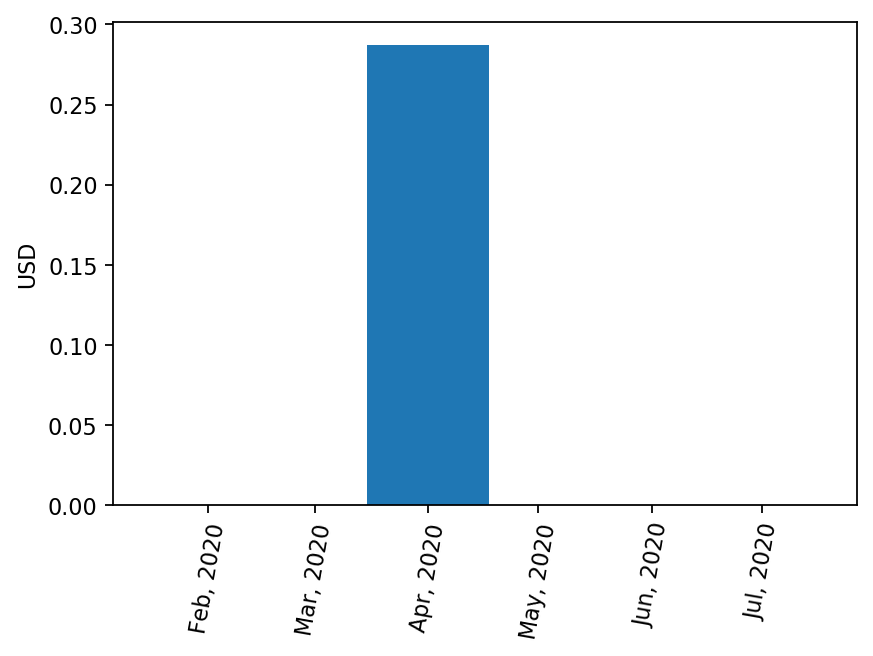
**איור 25 - גרף ההספק כתלות בזמן שנמדד לפי המונה החכם**



**איור 26 - גרף ההספק שהמערכת מזהה כתלות בזמן**

נשים לב ראשית שהגרף שהמערכת מציגה הרבה יותר חלק, והסיבה היא שהגדרנו סף שמתחתיו אנחנו לא מזהים מאורעות. הדבר בא לידי ביטוי במיוחד בכך שבחלק גדול מהדגימות ההספק בגרף הוא בין 0 ל0.5kw והמערכת שלנו מתעלמת מההספק הזה. אולם אפשר לראות שהמידע שאנחנו מציגים למשתמש מציג בצורה טובה את טיב האות שנדגם (ניתן לראות למשל בOven שאנחנו מצליחים להראות את כל העליות והירידות), ואת הגדול האבסולוטי שלו.

המידע שאני מנתחים כדי לבנות את הגרף הזה מאפשר לנו גם לבנות גרף מחיר שבו ניתן לבחור קבוצת מכשירים להציג את המחיר או את ההספק הכולל שלהם לפי חודשים:



**איור 26- גרף המחיר כתלות בזמן (קבענו שמחיר kwh הוא 0.1$( של קבוצת המכשירים מאיור 25**

ניתן לראות שהגרף מציג בצורה נוחה ופשוטה למשתמש את המחיר הכולל שהוא משלם עבור קבוצת מכשירים שהוא בוחר (עד כדי הסטייה של המערכת מההספק האמיתי), ובכך לשלוט בצורה טובה יותר על צריכת החשמל הביתית שלו.

# סיכום, מסקנות והצעות להמשך

## בחינת תוצאות הפרויקט מול המטרות שהוגדרו מלכתחילה

בתוכנית העבודה רשמנו:

*" תוצרי הפרויקט יהיו אלגוריתם ללמידה של מערכת חשמל, וממשק משתמש (ייתכן בצורת אפליקציה) המממש את האלגוריתם ומציג בצורה ברורה לצרכן את המכשירים המחוברים לרשת החשמל שלו וצריכתם.*

*כמו כן נבנה בסיס נתונים דינאמי עם מכשירי חשמל וצריכתם שיתעדכן לפי האלגוריתם שלנו ופלטים מהמשתמש.*

*בנוסף נציג תוצאות של סימולציות בהן האלגוריתם מצליח להבחין בשינוי ברשת החשמל ועדכון בסיס הנתונים. יתר על כן, נציג את הדיוק שהאלגוריתם פועל בו עבור אותן סימולציות- זיהוי תקין של המכשירים, צריכת חשמל נכונה עבור כל מוצר, וכו'..."*

ניתן לראות מהחלקים של מימוש התכנה וניתוח התוצאות שעמדנו במטרות שהצבנו לעצמנו, והצלנו לבנות מערכת כוללת שמבצעת את כל הדברים האלו, כאשר במהלך העבודה על הפרויקט חקרנו את הניסיונות הקיימים לפתרון בעיית ה – NILM ואת ביצועיהם, ולאחר מכן בנינו את האלגוריתם שלנו וממשק נוח להצגתו למשתמש, כאשר החלק של התוצאות פירטנו והראנו עד לאיזה דיוק הגענו.

## הצעות לשיפור ביצועי המערכת

ראשית על מנת לשפר את ביצועי המערכת, צריך לחשוב על שיטה חכמה יותר מהגדרת סף קבוע על מנת לזהות מאורע, למשל להגדיר סף שהערך שלו תהיה פונקציה של הרעש במדידות, ואז נוכל לזהות מאורעות בודדים במקרים לא רועשים, ובכך באופן ישיר לשפר את האלגוריתם.

נקודה נוספת שיכולה לשפר את האלגוריתם היא להוסיף למדד מאפיינים נוספים להשוואה בין מכשירים, למשל זמן הפעלה ממוצע או במידה ומדובר במונה אחר אפשר להשתמש במאפיינים חשמליים נוספים.

בנוסף, מתן האפשרות למשתמש להגדיר מראש מכשיר או לקשר באופן ידני מכשיר יכולה לעזור למערכת, אבל במחיר של ממשק משתמש פחות נוח.

## עבודה עתידית על הנושא ורעיונות לפרויקטי המשך

תחום ה – NILM הינו תחום עצום וישנו מקום רב להמשך עבודה בתחום בחלק זה ננסה לרשום כמה מהרעיונות שאפשר לבצע:

1. בשביל לשפר את תוצאות האלגוריתם אפשר לנסות לשלב בתוכו שיטות למידת מכונה שונות נוספות כמו השימוש ב – Reinforcement learning, מודלים מרקוביים חבויים, או פתרונות המשתמשים בלמידה עמוקה (פתרונות כמו CNN שהוצג בחלק של הסקירה הספרותית).
2. עבודה נוספת בתחום שיכולה להרחיב פרויקט זה תהיה להשתמש בנתונים שנאספו על ידי המונה והאלגוריתם בשביל לזהות חריגות במדידות של המונה ובכך לזהות תקלות עתידיות במכשירים ולהזהיר את המשתמש מהם. שימוש אחד של כזו תוספת גם בחלקים אחרים של הפרויקט יכול להיות שכאשר המודל מזהה שהמזגן בבית של המשתמש עומד להתקלקל הוא יתחיל להציע לו פרסומות לקניית מזגנים חדשים ולטכנאי מזגנים באפליקציה.

## קישור לעמוד הגיטהאב של הפרויקט שלנו:

<https://github.com/matan-matzliach/nilm_final_project>

מקורות

**מאמרים שבהם נעזרנו:**

[1] Jorge Revuelta Herrero, Alvaro Lozano Murciego, et al: Non Intrusive Monitoring (NILM): A State of the Art (2017): <https://www.researchgate.net/publication/318510754_Non_Intrusive_Load_Monitoring_NILM_A_State_of_the_Art>

[2] Jack Kelly and William Knottenblet: Neural NILM: Deep Neural Networks Applied to Energy Disaggregation (2015): <https://arxiv.org/pdf/1507.06594.pdf>

[3] GEORGE W. HART: Nonintrusive Appliance Load Monitoring (1992): <https://ieeexplore.ieee.org/document/192069>

[4] Alejandro Rodriguez-Silva and Stephan Makonin: Universal Non-Intrusive Load Monitoring Using Filter Pipelines, Probablistic Knapsack, and Labelled Partition Maps (2019): <https://www.researchgate.net/publication/334478365_Universal_Non-Intrusive_Load_Monitoring_UNILM_Using_Filter_Pipelines_Probabilistic_Knapsack_and_Labelled_Partition_Maps>

[5] M. Nguyen et al: A Novel Feature Extraction and Classification Algorithm Based on Power Components Using Single-point Monitoring for NILM (2015): <https://ieeexplore.ieee.org/document/7129156>

[6] Yuanmeg Zhang, Guanghua Yang, and Shaodan Ma: Non-intrusive Load Monitoring based on Convolutional Neural Network with differential Input (2019): <https://www.sciencedirect.com/science/article/pii/S2212827119307243>

[7] LI GAO, Bo YIN and Zhi-cheng ZHU: Load Identification of Non-intrusive Load-monitoring System Based on Time-frequency Analysis and PSO-SVM (2017): <https://www.researchgate.net/publication/316347079_Load_Identification_of_Non-intrusive_Load-monitoring_System_Based_on_Time-frequency_Analysis_and_PSO-SVM>