**המחלקה להנדסת תוכנה**

**פרויקט גמר –**

אגירה וסיווג של מסלולי גופים פיזיקליים באמצעות למידת מכונה  
Storage and Classification of Trajectories via Machine Learning Methods

**מאת**

**אביתר גולן, 203311733**

**מנחה אקדמי: דר' ראובן יגל אישור: תאריך:**

**אחראי תעשייתי: דר' שי אקו אישור: תאריך:**

**רכז הפרויקטים: דר' אסף שפנייר אישור: תאריך:**

מערכות ניהול הפרויקט:

|  |  |  |
| --- | --- | --- |
| # | מערכת | מיקום |
| 1 | מאגר קוד | <https://github.com/EvjaG/FinalProject> |
| 2 | קישור ליומן | <https://mail313575.monday.com/boards/2070942680/> |
| 3 | קישור לסרטון דוח אלפא | להקלטת מסך המחשב ודיבור מלווה, אפשר להיעזר במדריך והכלים במקור **[6] -**למשל screencast-o-matic, viewedit.com או <https://www.flashbackrecorder.com/express/>  <https://shotcut.org/> או באמצעים של תוכנות תקשורת כמו Google Hangout , Skype (היתרון של חלקם שנותנים שרות אחסון לסרטון ואפילו קוד להצגת הסרטון על גבי דף ההגשה). כמו כן מומלץ להתרשם מסרטוני המסירה ברשימת הפרויקטים משנה שעברה: בקישור המצורף:  [6] הדרכה על הקלטת סרטונים <http://the-openclass.org/core/item/30> וגם עם <https://www.flashbackrecorder.com/> |

|  |  |
| --- | --- |
| סוג הפרויקט | תעשייתי, חברת hi-tech |
| פרויקט מח"ר | לא |
| פרויקט ממשיך | לא |
| פרויקט זוגי: | לא |

כלי פיתוח

|  |  |
| --- | --- |
| שפות | Python, Java |
| DB | MongoDB |
| יצירת תרשימים לדוחות | StarUML |

תוכן עניינים

[0. מבוא 3](#_Toc92377489)

[1. תיאור הבעיה 3](#_Toc92377490)

[2. סקירת עבודות דומות והשוואה 3](#_Toc92377491)

[3. תיאור הפתרון 3](#_Toc92377492)

[דרישות ואפיון הפתרון 4](#_Toc92377493)

[האתגרים של הפתרון מבחינת הנדסת תוכנה 4](#_Toc92377494)

[המשך הפרק תיאור הפתרון 4](#_Toc92377495)

[מהי המערכת 5](#_Toc92377496)

[תהליכים ונתוני המערכת 5](#_Toc92377497)

[תיאור הכלים המשמשים לפתרון 5](#_Toc92377498)

[4. מה עשינו עד כה ? 5](#_Toc92377499)

[5. תכנית בדיקות 5](#_Toc92377500)

[בדיקות יחידה 5](#_Toc92377501)

[בדיקות אינטגרציה 5](#_Toc92377502)

[בדיקות מערכת 5](#_Toc92377503)

[6. נספחים 5](#_Toc92377504)

[א. טבלת סיכונים 5](#_Toc92377505)

[ב. רשימת\טבלת דרישות 5](#_Toc92377506)

[ג. תכנון הפרויקט 5](#_Toc92377507)

[ה. תרשימים 5](#_Toc92377508)

[a. תהליך הלמידה 5](#_Toc92377509)

[b. סיווג מחיישנים 5](#_Toc92377510)

[c. סיווג ממסד 5](#_Toc92377511)

[ו. רשימת ספרות \ ביבליוגרפיה 5](#_Toc92377512)

# מבוא

כיום מוערך שיש בעולם כ-47,840 כלי טיס צבאיים (WMMDA) בשימוש, יותר מ-4,500 לוויינים פעילים המרחפים במסלול מעל האטמוספירה ואין-ספור כלי נשק המסוגלים לנוע באוויר או בחלל.

במדינת ישראל ישנו צורך תמידי לשמור על בטחונה, לרבות בטחונה האווירי. במשך שנים רבות נאספו ותועדו מסלולי גופים רבים שנעו שתוך ומחוץ לשטחה האווירי של מדינת ישראל.

# תיאור הבעיה

הבעיה – המצב הקיים כיום: המעקב כיום הוא אחרי גופים ספציפיים בעזרת חיישנים שמוקצים בלעדית לכל גוף. הצפי בעתיד הקרוב הוא היכולת לעקוב אחרי גופים רבים שמסלולם אינו ידוע מראש, לרבות בניית מערכת אוטומטית למעקב אחר גופים, בעזרת חיישנים שלא יוקצו מראש לגופים מסוימים.

תיאור הבעיות מבחינת הנדסת תוכנה #TODO

# סקירת עבודות דומות והשוואה

נעשית בדיקה לגבי המערכות שבהם משתמשים בחברות לזיהוי וקטלוג מסלולים בצורה אוטומטית (לדוג' ברכבים אוטונומיים), הדורשת תשובה מצד הגורמים מהצד השני. לבינתיים אין תשובה, ולכן ביצעתי בהנחיית ראובן סקר שוק לגבי סוגי מסדי נתונים שיכולים להתאים לכמות גדולה של נתונים. השוואת מסדי הנתונים השונים נמצאת בנספחים. בנוסף , בהמשך המחקר נשווה בין סוגי למידה שונים, כולל למידה עמוקה ולמידת מכונה.

# תיאור הפתרון

## דרישות ואפיון הפתרון

הקמת מערכת הכוללת בסיס נתונים ומערכת לאגירת נתונים בזמן-אמת מחיישנים הקולטים גופים פיסיקליים הנעים במרחב, הכולל תיעוד מסלולם ותכונותיהם. המאגר יאחסן ויקטלג את הנתונים בעזרת הכלים של למידת מכונה ויאפשר זיהוי הגוף בהשוואה לגופים אחרים. נטמיע מס' אלגוריתמים של למידת מכונה ונעזר בהם על מנת למדוד את ביצועי המסלול של גוף חדש וסיווגו.

## האתגרים של הפתרון מבחינת הנדסת תוכנה

* הקמת מערכת מבוזרת שתוכל לאגור ולקטלג את המידע, ולמשוך אותו בשלב יותר מתקדם.
* בניית מערכת שתמדוד את ביצועי האלגוריתמים ותשווה ביניהם.
* בניית מחולל מסלולים לצורכי הזנת נתונים ולמידה מוקדמת.
* התאמה של נתונים מחיישנים וממאגרי מידע קיימים לצורה הניתנת ללמידה בעזרת למידת מכונה.
* בניית מערכת הלמידה באופן שבו מסלולים באורכים שונים יוכלו להתאים למערכת לומדת, היות ומס' הכניסות מוגבל לגודל מסויים.

הפתרון אליו הגענו מכיל מס' רכיבים, ובהם:

* מחולל מסלולים, לצורך יצירת מסלולים בסיסיים עליהם נלמד בהמשך (היות והמסלולים הקיימים במערכות של החיל הם חסויים) ומערכת הכותבת מסלולים אלו לקבצי תיעוד וקבצים ללמידה.
* מערכת למידה – יצריך ככל הנראה רשת נוירונים, לא בטוח עדיין אם CNN או ANN #TODO
* מסד נתונים – נוכל לאחסן בו את כל המסלולים והמידע שקשור ללמידת המכונה לצורך שליפה מהירה במידת הצורך (לדוג' לזיהוי אובייקט הנע במהירות במרחב)
* רק אחר כך (אפשר בתתי פרק) פירוט על אופן הפתרון עצמו - הטכנולוגיות שמשתמשים בהן, חלוקה למודולים, High Level Design, תרשים זרימה אם רלוונטי, פירוט של אלגוריתמים מוכרים שנעשה בהם שימוש (עם חלוקת קרדיט והפניה לביבליוגרפיה), פירוט אלגוריתמים יעודיים שהמצאתם\המצאתי\המצאנו (בלי חלוקת קרדיט) וכו'.

# המשך הפרק תיאור הפתרון

נא לפנות לנספחים לטובת תרשימי זרימה המתארים את פעולות המערכת המרכזיות.

במקרה שיש הרבה איורים יש להפנות לנספחים מטה (ראו שם).

פרק זה יכלול את כל התרשימים אשר למדתם -- תרשימי שימוש, ארכיטקטורת המערכת,

אלגוריתמים , תיאור הקוד ותיכון (תרשימי הפצה ומודולים עיקריים, שפות וכלים מתוכננים לשימוש כגון  מערכות הפעלה, שימוש ברכיבים קיימים, סביבות, כלי בדיקה).

דיאגרמת רכיבים \ הפצה

מעבר לזה –המרכיבים העיקריים לתיאור תת-פרויקט זה (כמו שלמדתם בקורס בהנדסת תוכנה).

1. Use-cases
2. Design (תיכון).

|  |
| --- |
| Use Cases כאמור מיצגים את צד הלקוח  Requirements מייצגות את צד המערכת  לעיתים הלקוח יודע מה הוא רוצה , לדוגמא - תמיכה בIOS גרסאות 9 ומעלה , או לא רוצה בכלל תמיכה הניידים אלא רק במחשב נייח ומערכת הפעלה Win 10, אלו דרישות מערכת    לכן לא לכל Use Cases יש דרישות ישירות  ההפרדה, כפי שלמדתם היא:   1. Use Cases לתואר הפעולה שהלקוח יעשה -- ולרב הפעולות האלו בסוף יהיו מסכים או חלקים ממסך 2. דרישות המערכת  -- שיותר כוללות מערכות הפעלה , סוגי ניידים ( אם בכלל יש תמיכה במובייל), גרסאות תוכנה , חיבורים  מסד נתונים ( בדגש של איזה ואיזו גרסאות )    צריך לוודא שכל דרישה היא גם ברת בדיקה |
|  |

## מהי המערכת

ראו את [התרשים בנספחים](#_רכיבי_מערכת) המתאר את מבנה המערכת באופן כללי

## תהליכים ונתוני המערכת

המערכת תכלול ארבעה מצבים עיקריים:

* + מנוחה: המערכת מחכה לפעולת משתמש או לתהליכים נכנסים
  + סיווג:
    - [סיווג של נתונים חדשים](#_סיווג_מחיישנים) - נתונים שייכנסו מתוך חיישנים חיצוניים יעברו סיווג בזמן אמת על ידי המכונה הלומדת
    - [סיווג של מקבצי נתונים](#_סיווג_ממסד) - נניח שיוכנסו נתונים לרשת ממקור חיצוני (לדוג' מנהל הוסיף מסלולים שנרכשו מחברה חיצונית) שלא נלמדו עדיין, אנו נבצע עליהם תהליך סיווג במידה ואין ידע על איזה סוג מסלולים אלו.
  + [למידה](#_תהליך_הלמידה): המערכת תיקח את כל הנתונים שיש להם סיווג חיצוני (כלומר ידוע איזה סוג מסלול זה ופרטים נוספים קיימים עליו במידת האפשר), ותבצע תהליך למידה מאפס, עם שאיפה לרמת דיוק שתיקבע ע"י מנהל המערכת. הלמידה תתבצע אוטומטית אחת לתקופה במידה ונוספו נתונים מתאימים למסד מאז הלמידה האחרונה, או ידנית ע"י מנהל מערכת.

## תיאור הכלים המשמשים לפתרון

בפרוייקט השתמשתי עד כה ב-Python לכתיבת הקוד, ובספריות החיצוניות הבאות:

|  |  |
| --- | --- |
| הספרייה | הסיבה שהשתמשתי |
| NumPy | חישוב ויצירת המסלולים, פונקציות מתמטיות נוחות |
| pandas | בדיקה של נכונות החישובים של המסלולים, חישוב מהירויות, הצגה נוחה יותר להדפסה על המסך |
| OpenCV | המרת מסלולים ממערכים לתמונות, לטובת בדיקה האם רשתות קונבולוציה יתאימו ללמידה של המסלולים |
| Matplotlib | המחשת המסלולים בעזרת אנימציה |

# מה עשינו עד כה ?

* יצירת מחולל מסלולים, כולל כתיבה לקבצי PNG וCSV
* יצירת מודול לקריאת הCSV ופרסור (לדו’' מציאת המהירות בכיוון X בזמן T)
* בחירת מסד נתונים לעבודה ראשונית
* חקר על סוגי אלגוריתמי הלמידה השונים לטובת הפרוייקט
* יצירת מכונה לומדת לסיווג המסלולים
* ביצוע ניסויים ראשוניים ללמידה, ללא בדיקה
* בדיקת המסלולים במחלקת test
* כתיבת מודול לאחסון ושליפת המסלולים ממסד הנתונים

# תכנית בדיקות

על המערכת יבוצעו שלוש סוגי בדיקות, אותן אתאר בפירוט בפרק הקרוב:

## בדיקות יחידה

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| מס' | תיאור | תנאי קדם | צעדים | דאטה | צפי | מציאות | עבר/נכשל |
| 1 | כתיבת מסלולים | התיקייה data צריכה להיות ריקה מתוכן | 1. בחירת מס' המסלולים (X) 2. בחירת Y נקודות למסלול 3. הרצה 4. בדיקה שיש X מסלולים, והחלוקה בין תיקיות train ו-test בערך 80-20 | אין | קבלת 0.8\*X קבצים בתיקייה train, 0.2\*X קבצים בתיקייה test,  כל קובץ עם Y שורות |  |  |
| 2 | אחסון נתונים במסד | המסד צריך להיות ריק טרם תחילת הבדיקה | 1. בחירה של איזשהו סט של דאטה אקראי 2. הכנסת הנתונים למסד 3. משיכת הנתונים בחזרה מהמסד 4. השוואה של הנתונים מול המקור לשגיאות | יווצר על-ידי המערכת, גודל וכמות יקבעו מראש ע"י הבודק | לא יהיו שגיאות |  |  |

בהמשך הדרך ייתכן כי נחליט שנכון לבנות מערכת קבצים מבוזרת (כדוג' Hadoop). אם כן,

נצטרך לבנות מערכת להזנת המסלולים לתוך המסד שישב על גבי מערכת הקבצים המבוזרת  
בצורה נוחה ומהירה, ולבדוק גם אותו.

## בדיקות אינטגרציה

* שליפה של מסלול חדש ממסד הנתונים ולמידה עליו, ללא מידע חדש מהחיישנים.
* אחסון מסלול שעבר למידה במערכת עם הסיווג שקיבל כתוצאה מהלמידה

## בדיקות מערכת

איזה בדיקות אפשר לעשות ? מה הם הבדיקות ? – חזרו לעיין בשיעור המתאים מהקורס הנדסת תוכנה

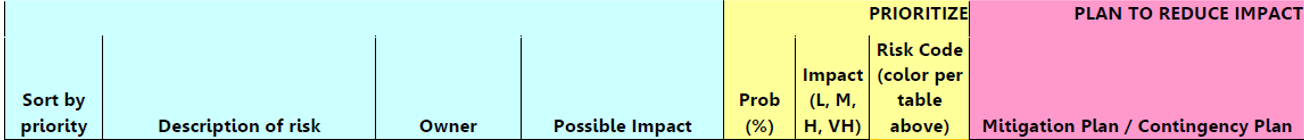
# נספחים

ספרות, תרשימים נוספים, תכנון הפרויקט, טבלת ניהול סיכונים, טבלת דרישות (URD),

**במקרה שלא היה מקום פרק "תיאור הפתרון" דאגרמות עם חשיבות משנית אפשר לשים בנספח במסכים (אם לא למעלה)**

## טבלת סיכונים

עדכון מצב הסיכונים– ע"פ המבנה הבאה



* עדכון רשימת המשימות והסיכונים – יש להשתמש בtemplate סיכונים מהקרוס "הנדסת תוכנה"

## רשימת\טבלת דרישות

- עדכון אם נוספו/בוטלו/השתנתה עדיפות וכדו'

פורמט טבלת הדרישות יהיה לפי המקובל בארגון. להלן דוגמא:

**טבלת דרישות (User Requirement Document)**

* **דרישות פונקציונליות**

|  |  |
| --- | --- |
| מס' דרישה | תיאור |
| 1 | המערכת שמקבלת את המסלולים צריכה להתאים למסלולים עם X נקודות (ייקבע על ידי הלקוח), ולדעת להתאים מסלולים עם יותר או פחות נקודות |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 | מסד נתונים לאגירת מסלול הגוף |

* **דרישות נוספות ולא פונקציונאליות**
  + **מהירות (זמן): המערכת צריכה לעלות תוך מס' שניות (לא יותר מ-5) מרגע זיהוי גוף ע"י חיישנים חיצוניים**
  + **עומס:**
  + **זמינות: רק בעת הצורך, המערכת לא תפעל תמיד**
  + **קיבול (צריכת זיכרון / CPU):**
  + **ניידות:**
  + **שימושיות:**
  + **תחזוקה:**
  + **אילוצי פלטפורמת מימוש:**

## תכנון הפרויקט

|  |  |
| --- | --- |
| 04.10.21 | <<פגישת הכרות עם הלקוח >> |
| 18.10.21 | תחילת למידה- מערכות מבוזרות לאחסון נתונים |
| 01.11.21 | המשך למידה |
| 15.11.21 | כתיבת דוח הצעת פרוייקט, תחילת תיכון אחסון המערכת |
| 29.11.21 | סיום תיכון ומעבר לעבודה על תיכון מערכת הסיווג |
| 13.12.21 | כתיבת מחולל מסלולים, בדיקת פורמטים לאחסון הנתונים |
| 27.12.21 | החלטה על הפורמט, בניית נתונים ללמידה עתידית |
| 10.01.22 | חקירת סוגי הלמידה למציאת למידה מתאימה |
| 24.01.22 | המשך חקירה |
| 07.02.22 | הגעה למסקנות ותחילת כתיבה של תוכנית עבודה למערכת |

1. סקר שוק
2. מסדי נתונים

בדיקת מערכות לאגירת מסלולים במערכות אחרות, לדוג' מערכות רכבים אוטונומיות, מטוסים וכו'.

בדיקת סוגי מסדי הנתונים השונים

לאחר התייעצות עם המנחה הוסכם כי שנינו מעדיף לעבוד עם מסד נתונים לא-יחסי, כלומר מסוג NoSQL. הבסיסים אותם אני בודק למען הפרוייקט הם MongoDB, Cassandra ו-HBase.

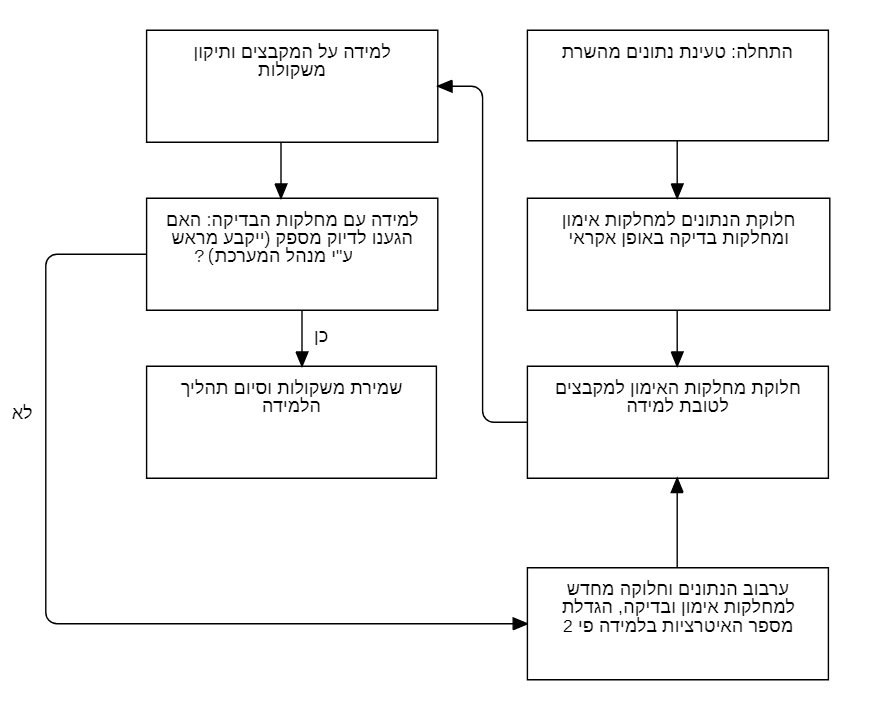
**Cassandra**בסיס נתונים מבוזר הכתוב בשפת ג'אווה, מיועד לטיפול בכמות נתונים גדולה העולה על מאות טרה-בייטים.  
הבסיס אינו משתמש במערכת Master-Slave אלא כל מחשב בצומת יוכל הפנות לכל צומת אחר. בנוסף, כל צומת מכיל שכפולים של מס' צמתים אחרים. המסד מבוסס שיטת "אחסון עמודה רחבה", כלומר משתמש בטבלאות עם שורות ועמודות, אך בניגוד מסדי SQL מאפשר ששמות ופורמטי העמודות ישתנו משורה לשורה.   
*חסרונות* הבסיס כוללים חוסר עקביות בעת שחזור מידע לצומת שנפלה, זמן קריאה ארוך מהשרת במידה והמפתח אינו ידוע למשתמש (לדוג' חיפוש של דוגמא ספציפית), וחוסר של תיעוד רשמי מאת הארגון המפתח, קרן Apache. שימושים ראשיים למסד כוללים תוכנות מסרים מיידים, מסחר אלקטרוני ועיבוד מידע מחיישנים בזמן אמת.

**HBase**בסיס נתונים הכתוב גם הוא בשפת ג'אווה, מבוסס על מערכת Bigtable של חברת גוגל, ומשתמש במספר תכונות שלו לרבות ביצוע פעולות על-גבי הזכרון בזמן אמת (לעומת גישה למידע הנמצא על גבי מחשבים רבים, פעולה הלוקחת הרבה יותר זמן), כיווץ מידע במוסד ושימוש במבני נתונים מבוססי Bloom Filter, אשר יכולים לקצר את זמן ההמתנה למשאבים מהדיסקים באופן משמעותי. בדומה ל-Cassandra, גם HBase מבוסס שיטת "אחסון עמודה רחבה". אחת מחוזקות המסד היא השימוש המובנה שלו ב-HDFS, מערכת הקבצים המבוזרת של Hadoop. המסד משתמש במערכת הקבצים HDFS של Hadoop, כך שאין צורך להשתמש במערכת MapReduce. חסרונות עיקריים של המסד כוללים שימוש במודל Master/Slave, דבר התורם לסיכוי לנקודת כשלון יחידה, אין ל-HBase שפת שאילתות מובנית (צריך להשתמש בממשק חיצוני, לדוג' Apache Hive), ותלות-יתר גבוהה בטכנולוגיות המובנות של Hadoop. שימושים שכיחים ל-HBase כוללים אנליטיקת לוגים מקוונת ואתרים מרובי תוכן (כגון נטפליקס, פינטרסט, פייסבוק).

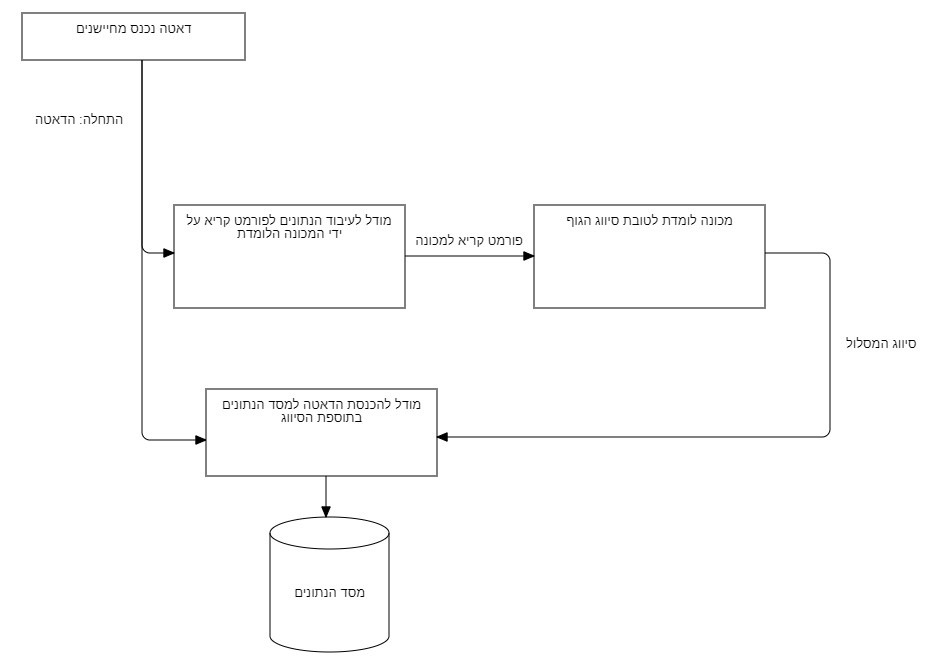
**MongoDB**בסיס נתונים הכתוב בשפות JS, C++ ופייתון, מבוסס שיטת אחסון מסמכים, המציע גרסת קהילה וגרסת עסקים הכוללת גישה מהירה וקלה לנתונים על גבי הרשת, ביקורת נתונים מובנית, אבטחת מידע והצפנה.  
MongoDB מאחסן את הנתונים על גבי המערכת כמסמכי JSON בינאריים, שיטה המאפשרת גישה מהירה למידע. גרסת העסקים גם כוללת זמינות גבוהה ע"י שכפול המידע ליצירת עודפות וריבוי צמתים לאחסון. חסרונות של המסד כוללות חוסר תמיכה רשמי לגרסת הקהילה (לדוג' תיקוני טלאי צריכים להתבצע באופן ידני), וגרסאות סקלביליות גוזלות יותר משאבי זכרון.

## תרשימים

## תהליך הלמידה

תיאור תהליך הלמידה של המערכת; יתבצע בהפעלה הראשונית ואחת לתקופה התיקבע מראש ע"י מנהל המערכת

## סיווג מחיישנים

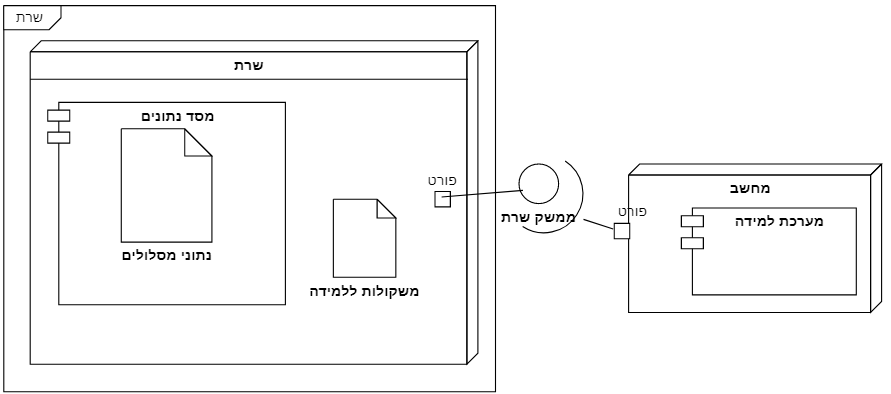
תהליך הסיווג של מסלול חדש שנקלט ע"י חיישנים חיצוניים

## סיווג ממסד

Diagram

Description automatically generatedתהליך הסיווג של מסלולים ממסד הנתונים שעוד לא סווגו

## רכיבי מערכת

תרשים המתאר את מבנה המערכת ממבט עליון

## 

## רשימת ספרות \ ביבליוגרפיה

עבור סקר השוק שבסעיף ד':

[**https://logz.io/blog/nosql-database-comparison/**](https://logz.io/blog/nosql-database-comparison/)[**https://db-engines.com/en/system/Cassandra%3BHBase%3BMongoDB**](https://db-engines.com/en/system/Cassandra%3BHBase%3BMongoDB)[**https://journalofbigdata.springeropen.com/articles/10.1186/s40537-016-0045-4**](https://journalofbigdata.springeropen.com/articles/10.1186/s40537-016-0045-4)