

## **הצעת פרויקט – עבודת גמר י"ד הנדסת תוכנה**

### **פרטי מגיש ההצעה**

**סמל מוסד:** 571281

**שם מכללה:** מכללת אורט סינגאלובסקי

**שם סטודנט:** קווין מנשרוב

**ת.ז:** 326969805

**שם פרויקט:** מערכת התראה לזיהוי אנומליות ברשת.

**מנחות פרויקט:** נילי נווה, אלי גוריאל, אפרת וינברג, אסף אמיר.

## תוכן עניינים

4		<b>תיאור הנושא</b>	
5		<b>רקע תיאורטי בתחום הפרויקט</b>	
5		זיהוי אנומליות ((Anomaly Detection	
5		למידת מכונה בלתי מפוקחת ((Unsupervised Machine Learning	
6		מכונת בולצמן מוגבלת ((RBM - Restricted Boltzmann Machines	
7		<b>תיאור הפרויקט</b>	
8		פירוט שלושת המודלים	
9		<b>הגדרת הבעיה האלגוריתמית</b>	
9		קלט ופלט המערכת	
10		תהליך הפתרון	
10		1. שלב הקלט - התמודדות עם ריבוי נתונים	
10		2. שלב העיבוד - ניתוח ושחזור מידע	
11		3. שלב ההחלטה (מידת פער)	
12		פירוט דאטה סטים	
12		דאטה סט CIC-IDS-2017 לזיהוי פעילות חשודה ברשת	
13		דאטה סט CIC-IDS-2017 לזיהוי ניסיונות חדירה	
14		דאטה סט CERT Insider Threat (למידה של פרופילי משתמש) לזיהוי הפרת מדיניות ארגונית	
15		<b>הליכים עיקריים בפתרון בעיה בטכנולוגיות הנדסה מתקדמות</b>	
15		איסוף וניתוח נתונים	
15		עבור מודל זיהוי חדירות	
15		עבור מודל זיהוי פעילות חשודה	
15		עבור מודל לזיהוי הפרת מדיניות	
16		ניקוי ונרמול נתונים	
16		הנדסת מאפיינים וחלונות זמן	
17		תרשים תהליכי מערכת	
18		<b>הליכים עיקריים בתחום למידת מכונה</b>	
18		1. עיבוד מקדים והכנת הנתונים	
18		2. תהליך האימון	
19		3. מנגנון הזיהוי לאחר שהמודל אומן	
19		4. קביעת סף ההחלטה ((Threshold	

<b>20</b>	<b>הליכים עיקריים בתחום רשתות מחשבים/תקשורת נתונים/אבטחת מידע</b>
20	1. ניטור והאזנה לתעבורה
20	2. חילוץ ופירסור
20	3. ניהול התראות
<b>21</b>	<b>תיאור פרוטוקולי תקשורת</b>
<b>21</b>	<b>פיתוחים עתידיים</b>
<b>22</b>	<b>תיאור טכנולוגיה הנדסה</b>
22	מנוע זיהוי וניתוח (שפת פיתוח Python)
24	שרת הניהול Backend (שפת פיתוח Java)
25	מסד הנתונים ((Database - MongoDB)
26	צד לקוח וממשק המשתמש (שפת פיתוח React)
27	ארכיטקטורת המערכת והתקשורת
<b>29</b>	<b>פרטים פורמליים</b>
29	לוחות זמנים
30	חתימת הסטודנט
30	חתימת רכז המגמה

## תיאור הנושא

התלות הגוברת של ארגונים במערכות ממוחשבות וברשתות תקשורת חשפה אותם בפני מגוון רחב של איומים. אחד האתגרים המרכזיים הוא לזהות פעילות חריגה או כוונת תקיפה בתוך אוסף רחב של נתונים, לפני שנגרם נזק.

נושא הפרויקט מתמקד בשימוש תחום ניתוח הנתונים ולמידת מכונה כמענה לאתגרי אבטחה אלו. כלים אלו מאפשרים מערכת הגנה שמסוגלת ללמוד התנהגות רגילה של הרשת, משתמשים ומערכות בארגון.

השימוש בניתוח נתונים ולמידת מכונה הוא לצורך זיהוי אנומליות (סטיות מהנורמה שנלמדה) בזמן אמת. למידת מכונה עוזרת לעבד כמות גדולה של נתונים (תעבורת רשת, פעילות משתמשים) ולמצוא דפוסים חשודים שקל לפספס.

היכולת לזהות אנומליות בעזרת למידת מכונה מתחלקת לכמה איומים. זיהוי ניסיונות חדירה - כגון סריקת פורטים, תנועה רוחבית חשודה ברשת שמטרת להגיע למערכות ארגון קריטיות, או למידע רגיש. איתור תקשורת זדונית - גילוי תקשורת חריגה ברשת המעידה על בקרת תוקף חיצונית או הוצאת נתונים רגישים, הנתונה בתוך התעבורה ברשת. אכיפת מדיניות ארגונית - זיהוי פעילות משתמשים או מערכות המפרה את מדיניות האבטחה הארגונית שנקבעה.

## רקע תיאורטי בתחום הפרויקט

### זיהוי אנומליות (Anomaly Detection)

אנומליה היא דפוס בנתונים שאינו תואם את ההתנהגות הצפויה ומוגדרת מראש. בהקשר לאבטחת מידע וסייבר זיהוי אנומליות הוא קריטי מכיוון שרוב מתקפות הסייבר מתחילות בפעולות חריגות שאינן מזוהות על ידי מערכות הגנה מסורתיות המבוססות על חתימות ידועות.

מערכות זיהוי אנומליות מבוססות על ההנחה שפעילות התקפית תהייה שונה לגמרי מפעילות לגיטימית רגילה. יתרון המרכזי הוא היכולת לזהות איומים חדשים ולא מוכרים, שכן אינן מסתמכות על ידע מוקדם על ההתקפה אלא על הכרת ה"נורמלי".

### למידת מכונה בלתי מפקחת (Unsupervised Machine Learning)

למידת מכונה היא תחום בבינה מלאכותית המאפשר למערכות ללמוד מנתונים ולשפר את ביצועיהן ללא תכנות מפורש. בפרויקט זה הגישה היא למידת בלתי מפקחת, שבה האלגוריתם מקבל נתונים גולמיים ללא תיוגים (ללא ידע מוקדם מהי התקפה ומה לא).

המטרה בלמידה בלתי מפקחת היא לגלות את המבנה הפנימי הנסתר של הנתונים. במקרה של זיהוי אנומליות, המודל לומד את המאפיינים של הנתונים ה"נורמליים" המהווים את הרוב המוחלט של המידע, ובכך מסוגלת לזהות כל נתון חדש שחורג ממבנה זה כחשוד.

### **מכונת בולצמן מוגבלת (RBM - Restricted Boltzmann Machines)**

RBM היא סוג של רשת עצבית מלאכותית המשתייכת למשפחת המודלים הגנרטיביים. היא מורכבת משתי שכבות - שכבה נראית (Visible Layer) המקבלת את הקלט, ושכבה נסתרת (Hidden Layer) הלומדת לייצג את המאפיינים הנסתרים של הנתונים. ההגבלה (Restriction) במודל זה היא שאין קשרים בכל שכבה, אלא רק קשרים דו כיווניים בין שתי השכבות. מבנה מוגבל זה מאפשר תהליך אימון יעיל ומהיר יותר (בהשוואה למכונת בולצמן מאלה). מטרתו הכללית של המודל היא ללמוד את התפלגות ההסתברות של הנתונים והייצוגים הנסתרים שלהם, והוא משמש לעיתים קרובות למשימות של למידת מאפיינים (Feature Learning), הפחתת ממדים ומערכות המלצה.

## **תיאור הפרויקט**

הפרויקט מתמקד בפיתוח ארכיטקטורה לזיהוי אנומליות ואיומים ברשת מבוססת למידת מכונה (נשתמש במכונת בולצמן) שתתריע בזמן אמת על פעילות חריגה ברשת.

בניגוד למודלי למידת מכונה אחרים (רגרסיה וסיווג) שחוזים תוצאה ישירות, מכונת בולצמן לומדת את הקשרים ואת המבנה הסטטיסטי בתוך הנתונים עצמם (איה תבניות יש בתוך הנתונים ואיך הם קשורים אחד לשני).

בהקשר לפרויקט, תפקיד המכונות הוא לא לסווג התקפות ידועות, אלה ללמוד את ההתפלגות ואת המבנה של נתונים "נורמליים" ברשת הארגונית. המערכת תאומן על תעבורת רשת כדי לבנות קו בסיס של התנהגות תקינה.

איומים ואנומליות יזוהו כאשר יגיע קלט חדש (כגון חבילת מידע) ואחד המודלים יתנו כפלט ציון גבוה או שגיאת שחזור גבוהה, מה שמסמן על כך שהנתון אינו תואם את הדפוסים הנורמליים שמערכת למדה.

### **פירוט שלושת המודלים**

המערכת תפעל על גבי שלושה מודלים מקבילים, ליבת הפרויקט היא פיתוח שלושה מודלים נפרדים של מכונות בולצמן, הפועלים בו זמנית.

### **המודל הראשון מזהה פעילות חשודה ברשת**

המודל יאומן על מאפיינים כלליים של תעבורת רשת (כגון נפחים, פרוטוקולים, תדירות ומשך התקשרויות). בכך המודל יצליח לזהות אנומליות סטטיסטיות רחבות בתעבורת הרשת כגון זיהוי דפוסים המעידים על תקשורת זדונית או סמויה (בקרת תוקף חיצונית - התקשרויות קטנות וקבועות ליעדים לא מוכרים, הוצאת נתונים - העברת נתונים חריגה בנפח ועוד).

### **המודל השני מזהה ניסיונות חדירה**

המודל יאומן על נתונים יותר ממוקדים, כגון יומני אימות, נתוני זרימה בין רכיבים פנימיים לצורך זיהוי התקפיים המכוונים לנכסי הארגון. מודל זה יזהה ניסיונות חדירה אקטיביים, לדוגמא - סריקות פורטים (ריבוי חיבורים כושלים מיעד בודד), התקפות כוח גס (ריבוי ניסיונות אימות כושלים), תנועה רוחבית חשודה ועוד.

### **המודל השלישי מזהה הפרות מדיניות ארגון**

המודל יאומן על דפוס גישה לגיטימיים של משתמשים וקבוצות למשאבים רגישים לצורך אכיפת כללי אבטחה. מודל זה יזהה פעולות המפרות את מדיניות הארגון לדוגמא - זיהוי מצב בארגון שבו חשבון ממחלקה מסוימת מנסה לגשת למשאב או מידע שלא ניגש אליו עד כה. פעולה זו תזוהה כאנומליה ביחס לפרופיל התנהגות הנורמטיבי של אותה קבוצת משתמשים.



## הגדרת הבעיה האלגוריתמית

הבעיה המרכזית שהמערכת צריכה לפתור היא: איך לזהות משהו "לא בסדר" \ "לא נורמלי" ברשת, מבלי שיודעים מראש איך נראית פעילות חריגה \ סטייה מהנורמה (המודל מסתמך על הנתונים שניתן לו שזה הנורמה \ פעילות לא חריגה).

במקום לחפש איומים ספציפיים, המודלים פועלים הפוך, למידת הנורמה:

כל שלושת המודלים לומדים ברמה גבוה ודיוק גבוה איך נראית שגרה תקינה, כל מה שלא תואם למודלים האלה (בין אם זה ניסיון פריצה, תקשורת חריגה או הפרת נהלים) יסומן כחריג.

## קלט ופלט המערכת

<u>הגדרה</u>	<u>פירוט</u>
קלט המערכת	מאפיינים רב ממדיים המייצגים את תמונת המצב של הרשת בכל רגע נתון. הוקטורים אינם מסומנים (למידה לא מפוקחת)
פלט המערכת	ציון שגיאה שחזור: ערך מספרי המציין את מידת החריגה. מתוך זה ניתן לבצע החלטה בינארית (נורמלי / לא נורמלי)

## תהליך הפתרון

### 1. שלב הקלט - התמודדות עם ריבוי נתונים

המערכת מקבלת בכל רגע נתון "תמונת מצב" של הרשת. האתגר הוא ש"תמונת המצב" הזו מורכבת מהמון פרטים קטנים: כתובת המקבל, כתובת השולח, גודל הקובץ, שעה, סוג פרוטוקול ועוד. המערכת צריכה לקחת את כל אוסף הפרטים הזה ולהפוך אותו לשורה אחת של נתונים שהמחשב יכול לעבד.

### 2. שלב העיבוד - ניתוח ושחזור מידע

זהו שלב במערכת המבוצע על ידי מודלי למידת מכונה (מכונות בולצמן - RBM). כדי להבין אם הנתונים תקינים, המודל מבצע תהליך דו שלבי -

1. המודל לוקח את הנתונים ומנסה לתמצת אותם (למידת התבנית של הנתונים).
2. לאחר מכן, המודל מנסה לשחזר את המידע המקורי מתוך אותו תמצות, בהתבסס על ה"שגרה" התקינה שהוא מאומן עליו.

אם הנתונים תקינים (מוכרים למערכת) - המודל יצליח לשחזר את הנתונים המקוריים בהצלחה, כי המודל "מכיר" את הדפוסים של הנתונים האלה.

אם הנתונים חריגים (פעילות לא מוכרת) - המודל יתקשה לשחזר את הנתונים המקוריים, והתוצאה לא תהייה מדויקת.

### 3. שלב ההחלטה (מדידת פער)

בשלב הסופי, המערכת צריכה לקבל החלטה של כן או לא (האם להפעיל התרעה?). המערכת עושה זאת על ידי חישוב של ההבדל של מה שנכנס לבין מה שהמודל הצליח לשחזר. הבדל זה נקרא "שגיאת השחזור".

שגיאה נמוכה: המערכת הצליחה לשחזר את המידע -> הפעילות תקינה.

שגיאה גבוהה: המערכת נכשלת בשחזור (הפער גדול מדי) -> הפעילות חשודה ומוגדרת כאנומליה.

המערכת משווה את גודל השגיאה ל"קו האדום" (סף) שנקבע מראש. אם השגיאה עוברת את הקו

האדום נשלחת התראה למנהל המערכת.

## פירוט דאטה סטים

### דאטה סט CIC-IDS-2017 לדיהוי פעילות חשודה ברשת

**משתנים בלתי תלויים** - מאפיינים סטטיסטיים של זרימה כגון: סך בתים / חבילות שנשלחו, יחס פרוטוקולים (TCP / UDP), משך חיבור, קצב העברת נתונים.

**משתנה תלוי** - ציון שגיאת שחזור (Reconstruction Error Score).

### דוגמא -

	A	B	C	D	E	F
1	Destination	Flow Durat	Total Fwd F	Total Back	Total Leng	Total Leng
2	3268	1.13E+08	32	16	6448	1152
3	389	1.13E+08	32	16	6448	5056
4	0	1.14E+08	545	0	0	0
5	5355	100126	22	0	616	0
6	0	54760	4	0	0	0
7	88	617	7	4	484	414
8	1031	8	1	1	6	6
9	88	881	9	4	656	3064
10	88	1056	9	6	3134	3048

### דאטה סט CIC-IDS-2017 לזיהוי ניסיונות חדירה

**משתנים בלתי תלויים** - מאפייני ניסיונות חדירה כגון: מספר ניסיונות אימות כושלים (Brute Force), מספר ניסיונות שנסרקו על ידי IP בודד (Port Scanning).

**משתנה תלוי** - ההחלטה אם הפעילות היא Intrusion / Normal.

**דוגמא -**

	AR	AS	AT	AU	AV
1	FIN Flag Co	SYN Flag C	RST Flag Co	PSH Flag C	ACK Flag C
2	0	1	0	0	1
3	0	1	0	0	1
4	0	0	0	0	0
5	0	0	0	0	0
6	0	0	0	0	0
7	0	0	0	1	0
8	0	0	0	0	1
9	0	0	0	1	0
10	0	0	0	1	0

**דאטה סט CERT Insider Threat (למידה של פרופילי משתמש) לזיהוי הפרת מדיניות ארגונית**

**משתנים בלתי תלויים** - מאפייני התנהגות משתמשים כגון: שעות התחברות חריגות, גישה לקבצים רגישים, שימוש בהתקנים חיצוניים, כמות מיילים שנשלחו מחוץ לרשת.

**משתנה תלוי** - ההחלטה אם היה הפרה של המדיניות או לא היה הפרה של המדיניות.

**דוגמא -**

	A	B	C	D	E
1	id	date	user	pc	activity
2	{S7A7-Y8Qz	#####	DTAA/RES0	PC-3736	Connect
3	{G7A8-G1O	#####	DTAA/BJC0	PC-2588	Connect
4	{R3L8-N0L\	#####	DTAA/EMZ0	PC-1479	Connect
5	{I2F1-B5FB	#####	DTAA/ZKH0	PC-1021	Connect
6	{P7R6-C5T\	#####	DTAA/RES0	PC-3736	Disconnect
7	{K5Q6-F1A0	#####	DTAA/CVW0	PC-0282	Connect
8	{M0F6-O2F	#####	DTAA/RQH0	PC-4225	Connect
9	{F5H8-O7Q	#####	DTAA/AQG0	PC-1127	Connect
10	{C2M6-A5G	#####	DTAA/OJH0	PC-1730	Connect

**קישור לדאטה סט CIC-IDS-2017:**

<https://www.unb.ca/cic/datasets/ids-2017.html>

**קישור לדאטה סט CERT Insider Threat:**

[https://kilthub.cmu.edu/articles/dataset/Insider\\_Threat\\_Test\\_Dataset/12841247?file=24855644](https://kilthub.cmu.edu/articles/dataset/Insider_Threat_Test_Dataset/12841247?file=24855644)

## הליכים עיקריים בפתרון בעיה בטכנולוגיות הנדסה מתקדמות

### איסוף וניתוח נתונים

מקורות הנתונים והאיסוף בשלב הראשון, עלינו להשיג מערך נתונים (dataset) המייצג תעבורת רשת אמיתית. מכיוון שהמערכת מבוססת על שלושה מודלים שונים, הנתונים צריכים להכיל מספר רבדים של מידע.

### עבור מודל זיהוי חדירות

נאסוף יומני אימות (Authentication Logs) משרתים ונקודות קצה (כגון SSH או Active Directory Logs) הכוללים מידע על ניסיונות כניסה מוצלחים וכושלים.

### עבור מודל זיהוי פעילות חשודה

נאסוף יומני זרימה (Network Flows) ויומני פרוטוקולים. המספקים מטא-דאטה על התעבורה (מי דיבר עם מי, מתי, כמה מידע הועבר).

### עבור מודל לזיהוי הפרת מדיניות

נשתמש בנתוני גישה למשאבים וקבצים (File Access Logs).

המטרה היא להשתמש במאגר נתונים קיים ומוכר או בנתונים מסימולציה מבוקרת, אשר מכילים בעיקר תעבורה תקינה לצורך אימון המודל, אך גם דוגמאות מתויגות של התקפות לצורך בדיקה של המערכת בשלב מאוחר יותר.

## ניקוי ונרמול נתונים

נתוני רשת מגיעים לרוב כטקסט לא מובנה, עם ערכים חסרים או "רעש" מיותר. בשלב זה נבצע ניקוי של רשומות פגומות והסרה של מידע לא רלוונטי ללמידה.

לאחר הניקוי נבצע תהליך של נרמול, מודלים מסוג RBM הם מודלים מבוססי "אנרגיה" משתמשים בפונקציות אקטיבציה יחסית רגישות. אם המודל יקבל שדה אחד עם ערכים קטנים ושדה אחד עם ערכים עצומים המודל יתקשה להתכנס.

לכן יש צורך לנרמל את כל הערכים המספריים לטווח אחיד כדי שהמשקולות ברשת ילמדו בצורה מאוזנת.

## הנדסת מאפיינים וחלונות זמן

שלב זה מתרכז בהתאמת הנתונים למודל ה-RBM. הפרויקט זה לא נזין את המודלים עם חבילות מידע (Packets) בודדות זו אחר זו, אלא נבצע תהליך של איחוד נתונים בחלונות זמן.

הסיבה לשלב זה נובעת משתי סיבות, אבטחה וביצועים:

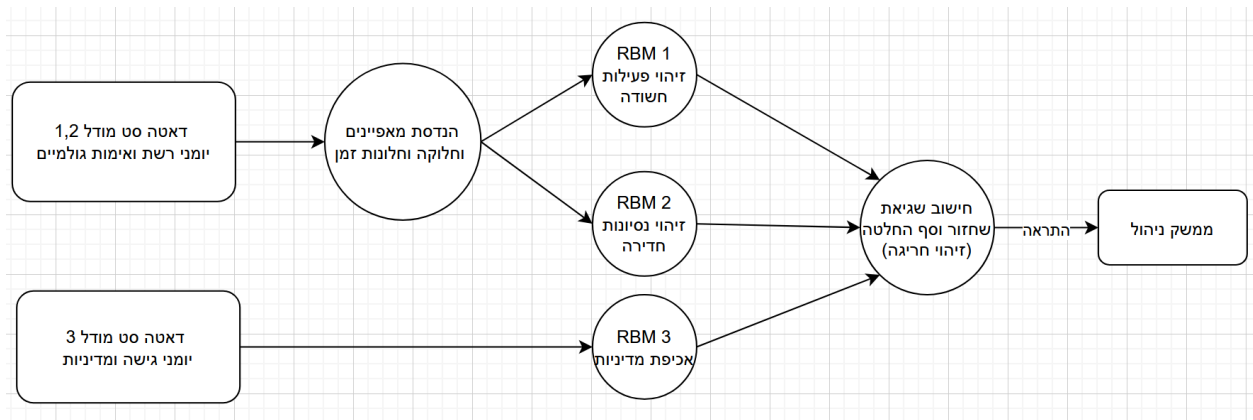
- 1. הקשר התנהגותי (Context) -** התקפות רבות, כגון סריקת פרוטים או מניעת שירות (DDOS), מורכבות מאלפי חבילות שכל אחת מהן נראית תקינה בפני עצמה. האנומליה מתגלה כאשר בוחנים את הקצב והכמות לאורך זמן, ניתוח חבילה בודדת תחסר את הקשר בין החבילות לכן הוא לא שימושי.
- 2. יעילות חישובית -** ברשתות מודרניות עוברות מיליוני חבילות בשנייה ברשת. ניסיון לעבד כל חבילה בנפרד ייצור עומס חישובי ועיכובים.

המערכת תחלק את התעבורה לחלונות זמן קבועים (לדוגמא, כל חמש שניות). עבור כל חלון זמן ניצור וקטור מאפיינים המסכם את הפעילות באותו רגע - כמה נסיונות אימות נכשלו ?, מה היה היחס בין חבילות שנשלחו לאלו שהתקבלו?

תהליך זה הופך את זרם הנתונים הגולמי לטבלה מסודרת של "מצבי רשת", המאפשרת למודל ה-RBM ללמוד את ההתפלגות הסטטיסטית של הרשת ולזהות חריגות.



## תרשים תהליכי מערכת



## הליכים עיקריים בתחום למידת מכונה

### 1. עיבוד מקדים והכנת הנתונים

לפני שהנתונים נכנסים למודל, עליהם לעבור התאמה מכיוון שמכונת בולצמן רגישות לטווחים שונים של מספרים.

נרמול - המרה של כל הערכים המספריים (כגון גודל חבילה, משך שיחה) לטווח באמצעות שיטת Min-Max Scaling. חלק זה חשוב כדי ששדות בעלי ערכים גדולים לא השתלטו על פונקציית האנרגיה של המודל.

קידוד קטגוריאל - המרה של נתונים טקסטואליים (כגון סוג פרוטוקול) לייצוג מספרי, כך שהרשת תוכל לעבד אותם.

חלוקת דאטה - הפרדת הנתונים לסט אימון המכיל תעבורה תקינה בלבד, וסט בדיקה המכיל ערבוב של תעבורה תקינה והתקפות, לצורך אימות הביצועים.

### 2. תהליך האימון

האימון מתבצע בשיטת למידה לא מפוקחת על שלושת המודלים במקביל.

המודל לומד את התפלגות ההסתברות של הנתונים התקינים. המודל מנסה למצוא את מערך המשקולות שיגרום לנתונים התקינים להיות בעלי אנרגיה מינימלית.

נשתמש באלגוריתם Contrastive Divergence. האלגוריתם מבצע קירוב של הגרדיאנט על ידי דגימה

(Gibbs Sampling) הוא מעביר את הנתונים מהשכבה הנראית לנסתרת ובחזרה, מעדכן את

המשקולות כדי לצמצם את ההפרש בין הקלט המקורי לשחזור שלו.

### 3. מנגנון הזיהוי לאחר שהמודל אומן

שחזור - כל וקטור נתונים חדש ( $V$ ) שנכנס למערכת עובר דרך הרשת. הרשת מייצרת ייצוג פנימי של וקטור הנתונים ( $H$ ) ומנסה לשחזר אותו בחזרה ( $V$ ).

חישוב השגיאה - המערכת מחשבת את המרחק המתמטי בין הקלט המקורי ( $V$ ) לבין הפלט שמשוחזר ( $V$ ). לצורך חישוב בשגיאה נשתמש בפונקציית Mean Squared Error.

רציונל - אם הרשת זיהתה דפוסים דומים באימון (התעבורה תקינה), השחזור יהיה מדויק והשגיאה תהייה קרובה לאפס. במקרה והרשת נתקלה בדפוס חדש (חשוד) היא תיכשל בשחזור והשגיאה תהייה גבוהה.

### 4. קביעת סף ההחלטה (Threshold)

הסף נקבע על סמך ביצועי המודל על סט הנתונים הנקי (ללא חבילות חשודות). לדוגמא נקבע את הסף כך שיכיל 95% אחוז מהשגיאות הנורמליות.

כל חריגה מעל הסף זה בזמן אמת תסווג מיד כאנומליה ותפעיל התראה.

## **הליכים עיקריים בתחום רשתות מחשבים/תקשורת נתונים/אבטחת מידע**

### **1. ניטור והאזנה לתעבורה**

המערכת פועלת כרכיב האזנה ברשת. כדי לקלוט את כל המידע, כרטיס הרשת (NIC) מוגדר לעבוד במצב Promiscuous Mode (מצב האזנה מלאה).  
במצב רגיל, כרטיס רשת מתעלם מחבילות שלא מיועדות אליו. במצב האזנה מלאה, הכרטיס קולט ומעביר למעבד את כל החבילות שעוברות. גם אם הן מיועדות למחשבים אחרים. דבר זה מאפשר למערכת לקבל תמונת מצב מלאה של התעבורה זמן אמת.

### **2. חילוץ ופירסור**

לאחר קליטת חבילות המידע הגולמיות (רצף של אחדים ואפסים) מתבצע תהליך של פירסור - פירוק החבילה לשכבות לפי מודל OSI.  
המערכת קוראת את הכותרות של הפרוטוקולים כדי לחלץ שדות מידע קריטיים (כגון כתובות IP מקור ויעד, פורטים, דגלים, גודל Payload).  
שלב זה הופך את המידע לנתונים מובנים שניתן לבצע עליהם חישובים סטטיסטיים.

### **3. ניהול התראות**

כאשר המודל מחשב שגיאת שחזור גבוהה החוצה את הסף המוגדר, המערכת מייצרת אירוע אבטחה. ההתראה נשלחת לממשק ניהול ונשמרת במסד הנתונים. היא תכיל זמן האירוע, מה האירוע שזוהה, הישויות המעורבות (כתובות IP).

## תיאור פרוטוקולי תקשורת

1. פרוטוקולי תעבורה (TCP / UDP) המערכת מנתחת לעומק את שכבת התעבורה (רמה 4 במודל OSI).  
TCP: המערכת עוקבת אחר תהליך לחיצת היד המשולשת (Three-Way Handshake). ריבוי של חבילות עם דגל SYN ללא מענה ACK תואם מעיד לרוב על סריקת פורטים או התקפת Dos.  
UDP: מכיוון שזהו פרוטוקול ללא חיבור (Connectionless), המערכת מנטרת נפחים חריגים שעשויים להעיד על הצפת מידע או ניסיונות אימות לשירותים כמו DNS.

2. פרוטוקולי בקרה (ICMP) פרוטוקול המשמש לבדיקות ואבחון (כמו Ping).  
האנומליה: תעבורת ICMP אמורה להיות דלילה מאוד. המערכת תזהה דפוסים של חבילות ICMP גדולות או תדירות גבוהה כחשד להגנבת מידע בתוך הפינג (ICMP Tunneling).

## פיתוחים עתידיים

1. מערכת מניעה אקטיבית: שדרוג המערכת כך שלא רק תתריע, אלא גם תבצע חסימה. המערכת תתממשק לחומת אש (Firewall) ותוסיף כלל חסימה לכתובת ה-IP החשודה באופן אוטומטי ברגע שהמודלים מזהים חריגה.
2. למידה היברידית: כיום המערכת מודיעה רק אם יש חריגה או לא. בעתיד, נוכל לשלב מודל מסווג, שירוך אחרי שהחריגה זוהתה, מודל זה ינסה לתת לחריגה שם ספציפי על סמך דוגמאות עבר. המטרה היא לתת לנהל הרשת מידע מדויק יותר על החריגה.

## תיאור טכנולוגיה הנדסה

פיתוח המערכת מבוסס על Stack טכנולוגי מודרני המורכב משלושה עמודי תווך: Python לטובת מודלי למידת מכונה ועיבוד התעבורה, Spring Boot לטובת צד השרת וניהול הנתונים, ו-React לבניית ממשק המשתמש לדשבורד.

### מנוע זיהוי וניתוח (שפת פיתוח Python)

צד זה של המערכת אחראי על האזנה לרשת, עיבוד מתמטי והרצת המודלים.

הוא יפותח בשפת Python בשל ספריית הכלים העשירה שלה בתחום הסייבר וניתוח נתונים.

**1. Scapy** - ספרייה מתקדמת למניפולציה של חבילות רשת, המאפשרת שליטה מלאה על שדות הכותרת (Headers) בכל שכבות ה-OSI.

Scapy משתמשת כחיישן של המערכת, היא מאזינה לכרטיס הרשת (NIC) במצב Promiscuous,

קולטת את התעבורה בזמן אמת, ומבצעת פירוק של החבילות כדי לחלץ נתונים גולמיים.

**2. Pandas & Numpy** - ספריית Numpy מספקת תשתית לחישוב מתמטי מהיר על מטריצות, ספריית Pandas מאפשרת ניהול ועיבוד מידע טבלאי וסדרות עתיות.

רכיבים אלו אחראים על הפיכת המידע הגולמי לקלט שניתן להזין למכונה:

אגרגציה: איחוד חבילות בודדות לחלונות זמן (לדוגמא, סיכום סטטיסטי של תעבורה כל 5 שניות)

נרמול: המרת הערכים בטווח באמצעות Min Max Scaling, תהליך קריטי עבור מודלים מבוססי

אנרגיה.

**3. PyTorch - ספריית למידה עמוקה (Deep Learning) המאפשרת בניית רשתות עצביות וחישוב מבוסס Tensors (מבנה נתונים לייצוג מספרים במימדים שונים).**

באמצעות ספרייה זו שלושת מודלי ה-RBM ימומשו. הספרייה אחראית על: ניהול הארכיטקטורה של הרשת (שכבות נסתרות ונראות), ביצוע תהליך השחזור (Forward and Back Pass), חישוב שגיאת השחזור בזמן אמת וזיהוי חריגה מהסף שנקבע.

## שרת הניהול Backend (שפת פיתוח Java)

צד זה אחראי על הלוגיקה העסקית. הוא אחראי על קבלת החלטות, ניהול תקשורת הא-סינכרונית מול המנוע של המודלים בלמידת מכונה, שמירת המידע והנגשתו למשתמש הקצה. השרת יפותח בשפת Java על גבי פריימוורק Spring Boot המוביל לפיתוח מערכות Enterprise יציבות.

**Spring Boot Framework:** נבחרה תשתית המאפשרת פיתוח מהיר של שירותי MicroServices. התשתית מספקת:

שרת Web מובנה: המאפשר הרצה עצמאית של האפליקציה ללא צורך בהתקנות חיצוניות מורכבות. ניהול תלויות (Dependency Injection): ארכיטקטורה המבטיחה קוד נקי, מודולרי וקל לתחזוקה ולבדיקה. אבטחה וביצועים: מנגנונים מובנים לטיפול בעומסים ואבטחת הקוד.

**תפקיד השרת והארכיטקטורה:** השרת ממומש בארכיטקטורת שכבות (Controller, Service, Repository) ומבצע מספר תפקידים:

- ממשק קליטת נתונים - חשיפת REST Controllers המאזינים לבקשות POST ממנוע הלמידת מכונה. שכבה זו אחראית על בדיקת תקינות הקלט כדי להבטיח שרק התראות במבנה JSON תקין נכנסות למערכת.
- לוגיקה עסקית - עיבוד התראות גולמיות שמגיעות מהמודלים.
  - הוספת חותמת זמן שרת וסטטוס התחלתי ("Open").
  - ניתוח ציון האנומליה (Anomaly Score), וקביעת רמת החומרה כדי לעזור למנהל השרת להתמקד בעיקר.
- שכבת הנתונים - שימוש בספריית Spring Data MongoDB המאפשרת מיפוי אובייקטים (ORM) ישירות למסמכי JSON. רכיב זה אחראי על שמירה יעילה ושליפה מהירה של היסטוריית ההתראות והלוגים ממסד הנתונים.
- ממשק לדשבורד: חשיפה נקודות קצה מסוג GET עבור צד הלקוח. השרת מבצע אגרגציה (ביצוע חישוב על קבוצת נתונים והחזרת ערך אחד) של הנתונים שולח סטטיסטיקות מוכנות כדי להקל על הדפדפן להאיץ את טעינת הדשבורד.



## מסד הנתונים (Database - MongoDB)

מסד נתונים מסוג NoSQL המבוסס על מסמכים (Documents-Oriented). מסד נתונים זה נבחר בשל הגמישות שלו בשמירת נתונים בפורמט JSON, התואם בדיוק את הפלט שמגיע ממודלי למידת המכונה. המערכת תנהל שלושה אוספי נתונים (Collections) עיקריים:

**1. אוסף התראות (Alert Collection):** זהו אוסף המרכזי במערכת, המרכז את כל אירועי האבטחה שזוהו. כל רשומה באוסף תכיל -

- Timestamp: חותמת זמן מדויקת של האירוע.
- source\_ip / dest\_ip: כתובות ה-IP המעורבות.
- detected\_by: שם המודל שזיהה את החרیגה.
- anomaly\_score: הציון המספרי של שגיאת השחזור (לאפשר סינון לפי חומרת השגיאה, לצורך תעדוף הטיפול בהתראות).
- status: סטטוס טיפול באירוע (פתוח/בטיפול/סגור) לשימוש מנהל המערכת.

**2. אוסף משתמשים והרשאות (User Collection):** משמש את מערכת הניהול לצורך בקרת גישה לדשבורד. כל רשומה באוסף תכיל -

- username: שם משתמש של מנהל הרשת.
- password: סיסמא של המשתמש (מוצפנת).
- role: רמת הרשאה (למשל - צופה בלבד או מנהל מלא).
- last\_login: תיעוד כניסה אחרונה למערכת.
- Created: מתי המשתמש נוצר.
- Modified: מתי שונה בפעם האחרונה.

**3. אוסף מטריקות וסטטיסטיקה (Metric Collection):** אוסף זה שומר נתונים רציפים לצורך הצגת גרפים בדשבורד, גם אם לא נוצרה התראה. נשמור דגימות זמן (Time-Series Data) של:

- ממוצע שגיאת שחזור בדקה האחרונה.
- כמות תעבורה כוללת. נתונים אלו מאפשרים לשרת ה-JAVA לשלוח היסטוריית ביצועים ולהציג למשתמש את בריאות הרשת לאורך זמן.

## צד לקוח וממשק המשתמש (שפת פיתוח React)

השכבה הוויזואלית המאפשרת למנהל הרשת לנטר את המצב.

**React js:** ספריית Javascript פופולרית לפיתוח ממשקי משתמש דינמיים רספונסיביים.

היא תשמש לבניית דשבורד ניהול:

הממשק יציג תרשים דינמי של שגיאת השחזור על ציר זמן. הגרף מאפשר לראות מתי שגיאת המודל

עולה מעל הסף המותר, דבר המעיד על אירוע אבטחה פעיל.

הצגת יומן אירועים (Event Log) אינטראקטיבי. הממשק כולל טבלה חכמה המאפשרת למנהל הרשת

לסנן, למיין ולחפש התראות לפי פרמטרים קריטיים (כגון: כתובת IP תוקפת, רמת חומרה, סוג המודל

שזיהה), ובכך מקלה על תחקור אירועי אבטחה.

שכבת ה-Frontend אחראית על ביצוע קריאות REST API א-סינכרוניות מול שרת ה-Spring Boot.

היא מושכת את הנתונים העדכניים ברקע ומעדכנת את המסך באופן מיידי ללא צורך בטעינה מחדש של

הדף, מה שמבטיח חווית ניטור רציפה וחלקה.

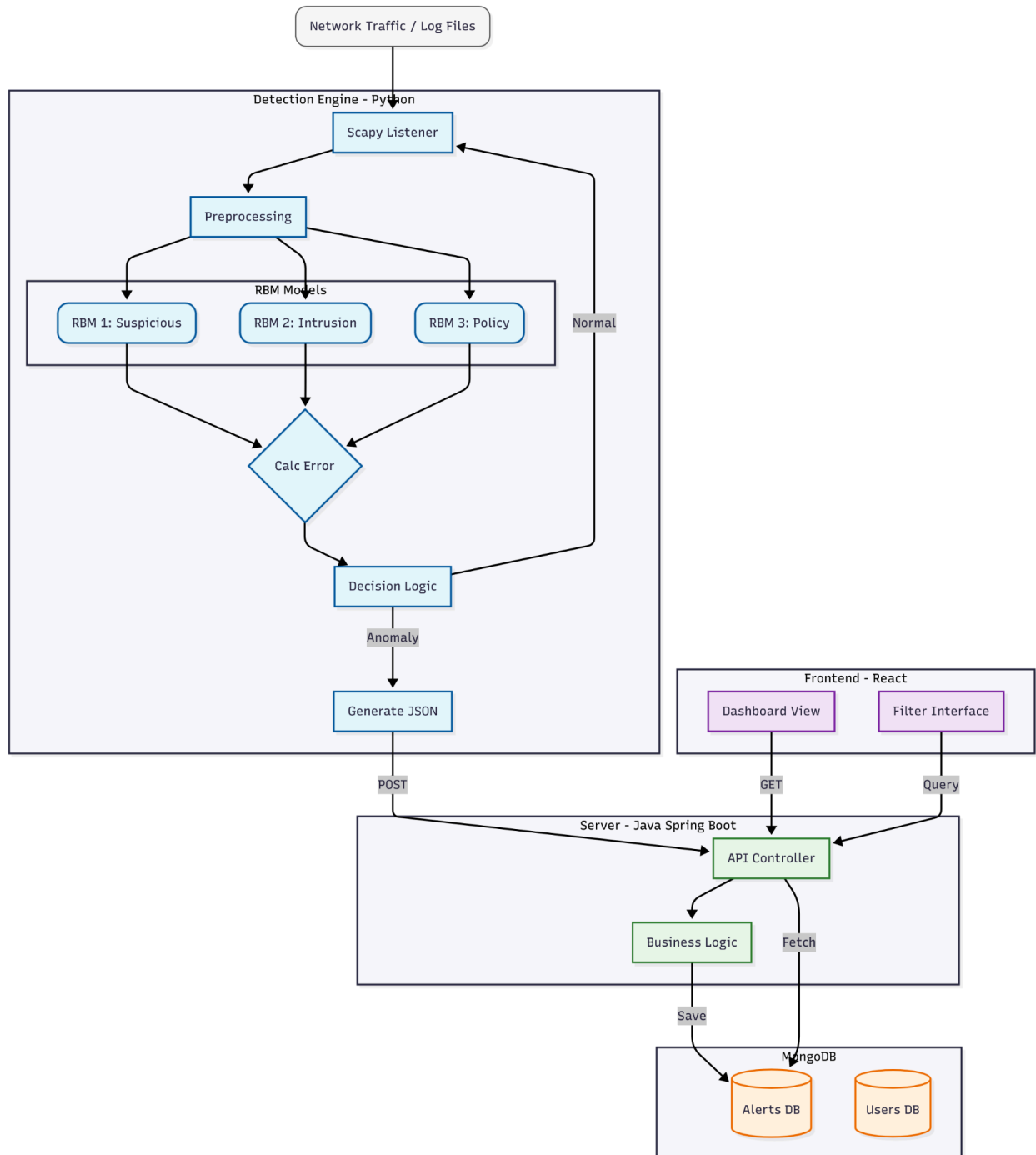
### ארכיטקטורת המערכת והתקשורת

המערכת מבצעת הפרדה בין מנוע הזיהוי לשרת הניהול. זרימת התקשורת בין שירות המזהה (Python) לשירות ה-Backend מתבצעת על גבי פרוטוקול HTTP באמצעות ממשק Restful API.

כאשר מודל ה-RBM ב-Python מזהה חריגה, הוא מייצר אובייקט **JSON** המכיל את פרטי האירוע (Timestamp, Source IP, Anomaly Score).

מנוע ה-Python שולח בקשת **HTTP POST** א-סינכרונית לכתובת ה-API של שרת ה-Spring Boot.

שרת ה-Spring Boot קולט את הבקשה, שומר את ההתראה ב-MongoDB ומעדכן את הדשבורד ב-React.



## פרטים פורמליים

### לוחות זמנים

<u>שלבי עבודה</u>	<u>תאריך סיום</u>	<u>פירוט</u>
חקר מקדים ולמידת טכנולוגיות	16.12.2025	למידה לעומק של מכונת RBM והבנת המתמטיקה שמאחוריו. חקר הנתונים כדי להבין אילו עמודות מיותרות ואילו לא.
הקמת והנדסת נתונים	24.12.2025	הורדת הדאטה סטים. כתיבת קוד לניקוי הדאטה, נרמול הדאטה.
פיתוח מודלי ה-RBM	15.1.2026	מימוש המודלים בעזרת PyTorch אימון שלושת המודלים יישום פונקציית שגיאת השחזור וקביעת הסף (Threshold)
פיתוח צד השרת	25.1.2026	הקמת פרויקט Spring Boot כתיבת הלוגיקה העסקית חיבור לבסיס הנתונים
ממשק המשתמש	10.1.2026	הקמת אפליקציית React ועיצוב הדשבורד. חיבור לשרת ה-Spring boot
אינטגרציה מלאה	28.1.2026	חיבור הקצוות של השרת והמודלים למידת מכונה. בדיקות המערכת וסימולציות.
תיעוד והגשה	28.1.2026	כתיבת ספר פרויקט והגשה.

**חתימת הסטודנט**

A handwritten signature in black ink on a light yellow background. The signature is stylized and appears to read "Keti".

**חתימת רכז המגמה**