



---

# בינה מלאכותית והגנת סייבר: הייפ ומציאות - חלק ד'

מאת [ד"ר יעקב רימר](#)

---

## הקדמה

בשנים האחרונות נוצר הייפ גדול סביב השינוי מהקצה עד הקצה שיביאו לעולם הגנת הסייבר שיטות של בינה מלאכותית (Artificial Intelligence - AI) בכלל ולמידת מכונה (Machine Learning - ML) בפרט. בסדרת מאמרים זו אני בוחן לעומק את הפוטנציאל של שיטות מתקדמות בלמידת מכונה לקידום משמעותי של יישומים שונים בתחום הגנת הסייבר, זאת בעקבות מאמר של ה-[Center for Security and Emerging Technology \(CSET\)](#) אותו אני סוקר.

[במאמר הראשון](#) הגדרתי את מסגרת הדיון והצגתי דיון לדוגמא שעסק בתרומה האפשרית של למידת מכונה לבדיקות חדירות. [במאמר השני](#) המשכתי בדיון במוצרי AV ו-EDR (יכולות ניטור) וביישומים לטובת כתיבה של קוד מאובטח. [במאמר השלישי](#) התמקדתי במערכות לגילוי חדירות (IDS) לרשתות ארגוניות. ובמאמר הזה, האחרון בסדרה, אדון בפוטנציאל של למידת מכונה לטובת אוטומציה לגילוי חולשות, ולטובת איסוף מודיעין ושיוך (Attribution) של תקיפות סייבר. לאחר מכן אחזור למסקנות המרכזיות של כותבי המאמר של CSET שהבאתי במאמר הראשון ואסכם את הדיון.

## למידת מכונה לטובת אוטומציה לגילוי חולשות

אחת המשימות הבסיסיות של שלב המוכנות והחוסן היא למצוא ולתקן חולשות קוד שעלולות לשמש את התוקפים. כלים אוטומטיים למציאת חולשות בקבצים בינאריים נחקרים כבר זמן רב, אולם רק בשנים האחרונות נעשו ניסיונות להשתמש גם בלמידת מכונה לטובת המטרה הזאת. אחת מאבני הדרך המשמעותיות בתחום זה היא תחרות ה-[Cyber Grand Challenge](#) של DARPA משנת 2016. מטרת התחרות הייתה לפתח מערכות אוטונומיות שיכולות לגלות ולתקן חולשות באופן אוטומטי. למרות שמספר צוותים בתחרות ניסו להשתמש בלמידת מכונה, כל התוכנות הזוכות השתמשו בשיטות אוטומציה לגילוי חולשות ותיקות יותר. הסתבר ששילוב נבון של פאזרים (Fuzzer) עם מנועי הרצה סימבולית (Symbolic Execution) מאפשר למצוא חולשות רבות באופן אוטומטי.

פאזרים הם כלי בדיקת תוכנות ותיק מאוד. הם מאפשרים להזין לתוכנית הנבדקת כמות גבוהה מאוד של קלטים שונים ומשונים. כך אפשר לבדוק כיצד התוכנית מתנהגת במידה ותקבל קלטים שאינם חוקיים, או שונים מהתבנית אותה היא תוכננה לקבל. כמובן שהפאזר גם מאפשר לנטר את ההתנהגות של התוכנית עבור כל סט כזה של קלטים. כך ניתן לאתר קלטים שגורמים לתופעות לא רצויות או קריסות של התוכנה, סימן היכר מובהק לבאגים או חולשות.

גם הרצה סימבולית היא טכנולוגיה ותיקה מאוד (בת כ-50 שנים) שנועדה במקור לאימות (Verification) של הנכונות הלוגית של קטע קוד מסוים. למרות שמה של השיטה, לא מריצים באמת את התוכנית, אלא עוקבים אחר התקדמות הפקודות ומבני הנתונים שלה. במהלך המעקב מייצגים את מבני הנתונים והמשתנים של התוכנית באמצעות סימבולים (להבדיל מערכים קונקרטיים), ואת המצב בו התוכנית נמצאת באמצעות תנאים לוגיים מתחום הלוגיקה המתמטית.

לא אתעמק יותר בטכנולוגיה הזו במסגרת המאמר הזה, אלא אסתפק באמירה כללית כי ביצוע מוצלח של הרצה סימבולית מאפשר לאמת לחלוטין את נכונות הקוד, בדומה להוכחה של משפט מתמטי. לכן ניתן לאתר באמצעותה עקרונית שגיאות או חולשות אבטחה.

עד כאן התאוריה. המציאות מורכבת משמעותית (!) יותר. לשתי השיטות האלו יש מגבלות רבות כאשר מריצים אותן על קוד של מערכות בעולם האמיתי. שוב אסתפק רק באמירה כללית שמסתבר ששילוב נבון שלהן, יחד עם שיטות של ניתוח תוכנה באמצעות מערכות חוקים (Static code analysis), מאפשר לאתר חולשות גם בתוכנות בעולם האמיתי. כפי שמעירים כותבי המאמר של CSET, העובדה שבאף אחד מהצוותים המנצחים בתחרות ב-2016 לא ראו צורך לשלב למידת מכונה במערכות שלהן, מעידה שלמידת מכונה רחוקה מלהיות הפתרון המועדף למציאה אוטומטית של חולשות.

האם דברים אלו השתנו בשש השנים האחרונות? לא מאוד. הדרך הטובה ביותר להעיד על כך היא קריאת דוחות הסיכום של חברות הגנת הסייבר השונות. אלו מדווחות על עליה בחולשות חדשות (כאלו שמכונות 0-day) בתקופה האחרונה.

למיטב ידיעתי אין עד היום פריצות דרך משמעותיות ביכולת למצוא חולשות בעזרת שיטות של למידת מכונה בלבד. יתכן ובגלל הקושי להכין מאגרי אימון לטובת [למידה מונחית \(Supervised learning\)](#) עבור בעיה זו ואולי בגלל האופן שבו חולשות מתבטאות בתוך הקוד. לדוגמה, חולשה יכולה להיווצר לא בגלל מה שכתוב בקוד, אלא דווקא מה שחסר בו. כגון חוסר באתחול של משתנה או בדיקת תקינות קלט חסרה או לקויה.

חוסרים מסוג זה קל יחסית לאתר אפילו באמצעות חוקי ניתוח קוד פשוטים (או קומפילרים), אבל זאת רק בתנאי שמיקום היווצרות הבעיה נמצא בסמיכות יחסית לקוד החסר. במידה ומדובר בבעיה שמתהווה לאורך זמן (ובין פונקציות שונות), קשה יותר לעלות עליה.

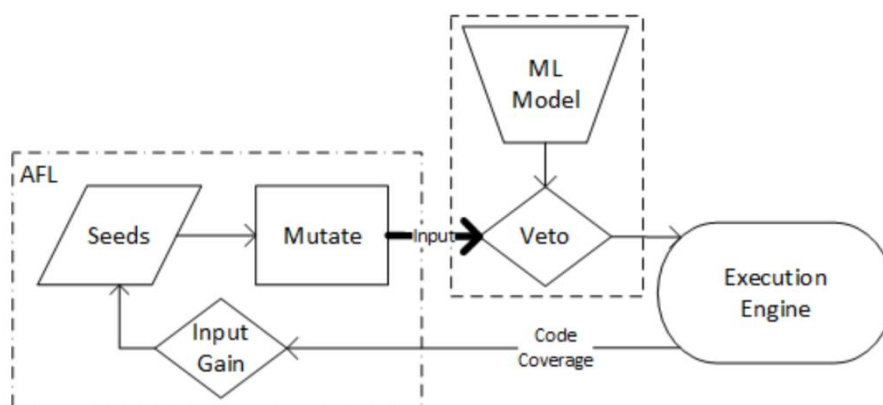
קיימים ניסיונות שונים ליעל את אופן בחירת הקלטים השונים עבור הפאזר. חלק ממחקרים אלו מדווחים על הישגים יפים. ומדוע זה חשוב? נניח ולתוכנית יש רק אפשרות לקלט אחד מסוג של מספר שלם

---

<sup>1</sup> בינה מלאכותית והגנת סייבר: הייפ ומציאות - חלק ד

(Integer). אם אנחנו רוצים לבדוק את כל האפשרויות, ומניחים שהוא ייקלט למשתנה בגודל 4 בתיים, אנחנו צריכים לעבור על יותר מ-4 מיליארד אפשרויות (וליתר דיוק,  $2^{32} - 1$ ). ואם יש לנו שני משתנים מסוג זה, נדרש לבדוק גם את כל הקומבינציות האפשריות של שניהם. מבינים לאן זה הולך? רק התחלנו. עכשיו נניח שקלט נוסף הוא קובץ, ואנחנו אפילו לא יודעים את אורכו. בקיצור, שיטות לבחירה נבונה של קלטים לפאזר הן נושא מחקר פורה וחשוב.

גישות אחרות מנסות לשלב [למידת חיזוקים \(Reinforcement Learning\)](#) יחד עם תהליך הפאזינג עצמו. לפחות בשלב זה לא ברור האם המאמץ אכן משתלם, או שמדובר במאמץ נוסף שהעלות-תועלת שבו לא ברורה. וגם אם כן, לא בטוח שהצלחה של הגישה הזאת בתוכנית אחת מעידה על היכולת להצליח גם עבור תוכניות אחרות. כדי להעמיק את ההסבר מדוע, נדרש הסבר ארוך על מבנה קבצים בינאריים וכיצד פאזרים עובדים, שחורג מאוד ממסגרת הדיון הנוכחי.



[דוגמה לסינון של קלטים עבור AFL באמצעות למידת מכונה, מקור: [arxiv](#)]

לסיכום, המחקר של שילוב למידת מכונה לטובת איתור חולשות אוטומטי הוא צעיר ועדיין קשה להעריך האם יביא לשינוי משמעותי בתחום. אבל גם אם נתבשר בקרוב על פריצת דרך, לא מדובר ככל הנראה בבשורה מהפכנית עבור עולם הגנת הסייבר. כזכור, יש גם תוקפים בתמונה וגם הם יודעים להפעיל את השיטות האלו ולהתעדכן בפריצות דרך. גרוע מכך, מגינים צריכים למצוא את כל החולשות בכל התוכנות שהם עושים בהם שימוש. לתוקף עשויה להספיק חולשה אחת בתוכנה אחת.

### למידת מכונה לטובת איסוף מודיעין ושייך (Attribution)

דוגמה לתרומה משמעותית של למידת מכונה לטובת הגנת סייבר היא מערכות לאיסוף וניתוח של מודיעין איומי סייבר (Cyber threat intelligence). כבר היום, שיטות של למידת מכונה ואינטליגנציה מלאכותית, ביחוד מתחום של [עיבוד שפה טבעית \(Natural Language Processing\)](#), מסייעות לאסוף מודיעין מתוך הרשת האפלה (Dark Web) או אתרים אחרים. שיטות נוספות של [ניתוח טקסט](#) מאפשרות לסווג ולנתח

<sup>1</sup> בינה מלאכותית והגנת סייבר: הייפ ומציאות - חלק ד

פוסטים או שיח בשווקים לסחר בחולשות קוד. מערכות אחרות מאפשרות לחברות לאתר ולנתח באופן אוטומטי וקטורי תקיפה שתוקפים עלולים לנצל לתקוף אותן.

מדובר במידע גלוי שמפורסם אודות החברה, מזהים של העובדים שלה, מהם המוצרים בשימוש שלה ועוד. כך ניתן, לפחות עקרונית, לשפר את המוכנות של החברה לתקיפת סייבר. בתחום הזה הכוח המלא של יכולות בינה מלאכותית בא לידי ביטוי ברור, וככל הנראה נראה התפתחויות נוספות בעתיד הקרוב. הבעיה שוב שגם תוקפים יודעים זאת. הם יכולים להשתמש בדיוק באותם כלים בשלב המודיעין המקדים (Recon) לקראת התקיפה שלהם.

יישום פוטנציאלי נוסף של למידת מכונה הוא היכולת לבצע שיוך (Attribution) של תקיפת סייבר. שיוך עשוי לאפשר תגובה מתאימה יותר לתקיפה, או לפחות לסייע להבין טוב יותר את הכוונות והמניעים שלה. כפי שמציינים כותבי המאמר, שיוך של תקיפה דורש פעמים רבות מיומנות גבוהה מאוד של חקירה וניתוח ממצאי התקיפה, מה גם שלפעמים הוא מסתמך בנוסף על מידע גאופוליטי או תוצאות מחקר אנושי של קבוצות תקיפה. לפיכך, כפי שהם מציינים, מדובר ברמת ניתוח שלמידת מכונה מתקשה לבצע ולכן לא נראה שהיא תוכל לבצע שיוך באופן מלא גם בעתיד.

מצד שני, כותבי המאמר של CSET מביאים דוגמאות ליישומים מוצלחים בתחום. ניתן לאתר תבניות של תקיפות דומות באמצעות שיטות [אשכול \(Clustering\)](#). ניתן להיעזר בתהליך השיוך במודיעין שנאסף אוטומטית מהרשת לגבי שיוך של תקיפות קודמות, בדומה לאיסוף מודיעין איומי סייבר. ואפשר לשייך פוגענים שמעורבים בתקיפה באמצעות זיהוי סגנון הקידוד של מפתחי הפוגען, באופן דומה [לזיהוי כותב של טקסטים בשפה טבעית \(Authorship attribution\)](#). אמנם המלאכה הראשית של השיוך תישאר בידי מומחי סייבר בשר ודם, אבל ניתן להקל על עבודתם באמצעות אוטומציה.

## סיכום הסדרה

בארבעת המאמרים בסדרה סקרתי מספר יישומי סייבר מתחומים שונים ונראה כי הגיע הזמן לסכם. אציין שבמאמר של CSET נסקרים יישומי הגנת סייבר נוספים מתחום ההגנה האקטיבית וההונאה (כגון מלכודות דבש דינמיות). אולם בחרתי לא להתעמק בהם והמתעניינים מוזמנים לקרוא עליהם במאמר שלהם.

כותבי המאמר סכמו את המסקנות המרכזיות שלהם במספר נקודות, אותן הבאתי כבר בפתיחת [המאמר הראשון](#). אחזור כעת לנקודות האלו (מובאות להלן בגופן מודגש) ואבחן אותם שוב לאור מוצרי הגנת הסייבר שסקרתי.

**נקודה ראשונה:** למידת מכונה יכולה לסייע למגינים לאתר התקפות סייבר בצורה מדויקת יותר. במקרים רבים לא מדובר בגישות חדשניות, אלא בהרחבה של אותם הדברים שכבר נעשים. כפי שכותבי המאמר של CSET מגדירים את זה, עלינו לצפות שהתועלת מלמידת מכונה תהיה אינקרמנטלית, ולא מהפכנית. הדגמתי את הנקודה הזו בדיון [במאמר הקודם](#) על מוצרי IDS. מוצרים אלו עדיין מבוססים בעיקר על חוקים וכנראה יישארו כך בעתיד הנראה לעין. רשתות מסוג GAN יסייעו כנראה לשפר את מוצרי ה-IDS, אבל לא יהוו שינוי מהפכני.

דברים דומים ראינו גם בדיון על [מערכות לזיהוי פוגענים](#). המעבר של מוצרי AV ו-EDR לענן מאפשר מחקר ML איכותי יותר וצפוי לשפר את היכולת הכללית לאתר פוגענים. אבל שוב, ככל הנראה מדובר בתוספת לשיטות קיימות, לא כתחליף להן. גם בדיון לעיל על שיטות לאיתור חולשות ראינו שהתרומה המרכזית של למידת מכונה (לפחות כרגע) הוא בייעול של אופן השימוש בפאזרים, לא בהחלפתם. אפילו בתחום מערכות לאיסוף וניתוח של מודיעין על איומי סייבר, בו התרומה של למידת מכונה היא משמעותית יותר, מדובר בייעול ויישום אוטומטי של עבודת מודיעין שאנשים ידעו לבצע גם קודם.

נעבור לסיפא של הנקודה הראשונה: **כמובן, אסור לשכוח שעצם השימוש בלמידת מכונה מוסיף משטחי תקיפה נוספים**. לאלגוריתמי ML יש פגיעויות מבוניות משלהם ([Adversarial AI](#)). מסתבר שקל יחסית לשתות במודלי סיווג שנוצרים באמצעות [למידה עמוקה](#) או שיטות [סיווג אחרות](#). נגעתי בנקודה הזו מספר פעמים במאמרים האחרונים, ומי שמעוניין להבין קצת יותר במה דברים אמורים, מוזמן לעיין בהסברים פשוטים לנושא שפרסמתי בעבר בכתבות שלי בדה-מרקר בלינק [הנה](#) ו[הנה](#).

**נקודה שנייה:** בעולם הגנת הסייבר קיימות משימות שגרתיות ומייגעות רבות. למידת מכונה עשויה לסייע לבצע חלק מהן בצורה אוטומטית ובכך לפנות את הזמן והקשב של המגינים. בחלק מהתחומים עדיין נדרשות פריצות דרך משמעותיות אל מול השיטות שקיימות כיום.

נפתח במשפט השני בנקודה הזו. לאורך כל הדיון הצגתי שוב ושוב את עובדות החיים. נכון להיום למידת מכונה עדיין אינה מספקת את "הסחורה", לפחות לא כפי שהיא מוצגת על ידי מחלקות שיווק של חברות סייבר שונות. כפי שנאמר לעיל, בד"כ מדובר בשיפור מסוים על השיטות המסורתיות, ובחלק מהדוגמאות שהבאתי (כמו מערכות אוטומציה של בדיקות חדירות) היא פשוט לא עובדת.

האם המצב הזה יישאר לנצח? כידוע הנבואה ניתנה לשוטים, וגם never say never. ברם, כבר היום ברור שידרשו פריצות דרך משמעותיות אל מול השיטות הקיימות. מצד שני, המשפט הראשון קובע כי למידת מכונה יכולה לסייע לפתור משימות מייגעות של מגינים. למשל על ידי ייעול משמעותי של תהליך איסוף מודיעין איומי הסייבר שנדון לעיל, או דרך סיוע בתהליך השייך של תקיפת סייבר.

**נקודה שלישית:** כניסה של שיטות למידת מכונה נוספות ישנו את "שדה הקרב" של הגנת הסייבר. לעיתים לטובת התוקפים ולעיתים לטובת המגינים. אבל ללא פריצות דרך נוספות, לא צפוי ששיטות

---

<sup>1</sup> בינה מלאכותית והגנת סייבר: הייפ ומציאות - חלק ד



למידה מכונה ישנו באופן דרמטי את תחום הגנת הסייבר. ראינו בכל היישומים שסקרתי בסדרה כי תרומתה של למידת המכונה בשלב הזה אינה משמעותית למניעה של התקפות סייבר. בנוסף, בכל מקום שיש שיפור לטובת המגינים, גם התוקפים יכולים לעשות שימוש בלמידת מכונה.

אם זה לטובת מציאת חולשות אוטומטית או איסוף מודיעין על חברה, עליהם דנתי במאמר הזה. או באמצעות [ייעול מוצר ה-IDS](#) באמצעות רשתות GAN, שיכולות לשמש את התוקפים למצוא שיטות חדשות לעקוף את אותם המוצרים ממש. וכפי שנאמר מספר פעמים, העובדות היבשות מלמדות שלמרות שנעשה שימוש של יותר משלשה עשורים בשיטות של למידת מכונה, מספר תקיפות הסייבר אינו יורד, אלא להיפך.

אז מה השורה התחתונה? לאור מצב שדה-הקרב של הסייבר, אין מנוס אלא להמשיך ולפתח יישומים חדשים של למידת מכונה לטובת הגנת סייבר. כן, גם אם בפועל אלו רק יאפשרו לנו "לרוץ הכי מהר שאנחנו יכולים, כדי להישאר באותו המקום".

## על הכותב

[ד"ר יעקב רימר](#) הוא יועץ בכיר ומרצה בנושאי סייבר, בינה מלאכותית וביולוגיה. יש לו תואר שני בלמידת מכונה ודוקטורט באימונולוגיה, שניהם ממכון ויצמן למדע. הוא עוסק במחקר מדעי באקדמיה במקביל ליעוץ במשרדי ממשלה ולחברות היי-טק. בעבר שימש בתפקידים בכירים בהיי-טק ובמשרד ראש הממשלה. בנוסף, הוא מלמד באוניברסיטת תל-אביב את הקורסים "בינה מלאכותית ויישומיה לביטחון", "בינה מלאכותית בעידן הסייבר" ו"מבוא להגנת סייבר" במסגרת תוכניות לתואר שני.

[קישור ללינקדאין](mailto:MrBigDataThemarker@gmail.com). מייל לתגובות: [MrBigDataThemarker@gmail.com](mailto:MrBigDataThemarker@gmail.com)

## מקורות לקריאה נוספת

Machine Learning and Cybersecurity - Hype and Reality. Micah Musser and Ashton Garriott. June 2021. <https://cset.georgetown.edu/publication/machine-learning-and-cybersecurity/>