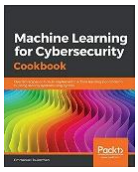


Machine Learning for Cybersecurity Cookbook
סיכום מתכונים

פרק 1 – Machine Learning for Cybersecurity

קודים של הפרק: https://github.com/PacktPublishing/Machine-Learning-for-Cybersecurity-Cookbook/tree/master/Chapter01

פיצול לאימון (train), אימות (validation) ומבחן (test)

לרוב בבעיות של ML, נרצה לפצל את המידע שלנו לשלושה חלקים: train, validation, ו־test. ה־train ישמש לאימון המודל, ה־validation ישמש ל"הכנה למבחן" של המודל, וה־test יהיה המבחן והערכת התוצאות הסופית.

קוד נמצא כאן

סטנדרטיזציה (standardizing) של המידע

על מנת לקבל מודל מהיר ויעיל מבחינת המחשב, נרצה לבצע סטנדרטיזציה של המידע. כלומר להפוך את כל הערכים שיהיו ממוצע 0, ושהשונות תהיה 1.

קוד נמצא כאן

סיכום (summarizing) נתונים עם PCA

במידה ויש לנו סט עם המון מאפיינים (features), מודל רב־ממדים, נרצה למחוק מאפיינים שלא יועילו למודל שלנו (לאו דווקא יזיקו לו). נעשה זאת באמצעות PCA.

קוד נמצא כאן

הפקת (generating) טקסט באמצעות Markov chains

נוכל להפיק וליצור טקסט באמצעות סט מידע גדול של כל מיני טקסטים. לדוגמה תגובות מזויפות שייווצרו לפי סט מידע של תגובות באתר מסוים.

ניתן לשנות את הפרמטר state_size, שמייצג מעברים בין זוגות מילים רצופות. אם נרצה משפטים מציאותיים יותר נגדיל את הפרמטר, אך זה יבוא על חשבון המקוריות של המשפט (כלומר התגובה תהיה דומה יותר לתגובות קיימות). אם נרצה משפטים פחות מציאותיים/תחבירים, המשפט יהיה חדש ומקורי יותר.

קוד נמצא כאן

ביצוע clustering עם scikit-learn

Clustering הוא אוסף של אלגוריתמים ללמידה לא מפוקחת (unsupervised learning), בהם סט המידע מתחלק לתוויות לפי דמיון/חלוקה לקבוצות. Clustering מאוד שימושי ב־cybersecurity להבחנה בין פעילות רשת נורמלית לבין רשת חריגה, ועזרה לסיווג תוכנות זדוניות ל"משפחות" (תוכנות זדוניות בעלי קוד דומה).

קוד נמצא כאן

אימון XGBoost classifier

Gradient Boosting (GBoost) הוא אלגוריתם הנחשב לאמין והמדויק ביותר לבעיות כלליות של ML. אנחנו נשתמש ב־XGBoost ליצירת גלאי תוכנות זדוניות במתכונים הבאים בספר.

קוד נמצא כאן

ניתוח סדרות זמן (time series) באמצעות הספרייה statsmodels

סדרת זמן היא סדרה של ערכים המתקבלים בזמנים רצופים/קבועים. דוגמה לכך היא מחיר בורסה שנדגמת מדי דקה בדיוק. בתחום ה־cybersecurity סדרת זמן הוא כלי שימושי לחיזוי תקיפות כמו קבוצת האקרים שמתכוננת למתקפה הבאה שלה.

קוד נמצא כאן

זיהוי חריגות עם Insolation Forest

זיהוי חריגות הוא זיהוי מקרים בסט מידע שלא תואם את הצפוי. ניתן לזהות כך פעילויות בלתי צפויות באפליקציות שיעידו לרוב על הונאה או משהו לא סביר. כדי לזהות חריגות כאלו נשתמש ב־ Insolation Forest.

קוד נמצא כאן

עיבוד שפה טבעית (natural language processing) באמצעות hashing vectorizer ו־tf-idf עם scikit-learn

לפעמים אנחנו רוצים לנתח אובייקטים טקסטואליים (ציוצים, משפטים, הודעות וכד'), ומכיוון שהאלגוריתמים דורשים קלט מספרי, אנחנו זקוקים להמיר טקסט למאפיינים מספריים.

טוקן (token) הוא יחידת טקסט (אפשר להגדיר שהטוקן יהיה משפט, מילה, אות וכד'). Count vectorizer לוקח קלט טקסטורלי ופולט וקטור המורכב מספירות הטוקנים הטקסטואליים. Hashing vectorizer הוא וריאציה של ה־CV, במהירות יותר גבוהה על חשבון הפרשנות. נותנים לכל מיני מילים משקלים שונים (לדוגמה מילים נפוצות כמו a ומילים נדירות). לצורך כך משתמשים ב־tf-idf.

קוד נמצא כאן

Hyperparameter tuning באמצעות scikit-optimize

Hyperparameter הוא פרמטר שערכו מוגדר לפני אימון המודל, שעוזר לדיוק ושיפור המודל (לדוגמה פרמטר האחראי על מהו ה־k ב־KNN). Grid Search היא דרך מעולה ונפוצה לכוונון הפרמטרים באופן קל ואוטומטי.

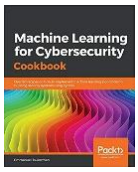
בשיטה זו, מגדירים טווח/רשימה של ערכים לכל/לחלק מהפרמטרים של המודל, וה־Grid Search ירוץ על כולם עד שהוא ימצא את הקומבינציה האופטימלית. חיסרון גדול הוא האינטנסיביות שבשיטה, המודל צריך להתאמן המון פעמים כדי לנסות פרמטרים שונים.

לעומת זאת, Bayesian optimization שנמצאת ב־scikit-optimize לא רצה על כל הפרמטרים כמו ב־ Grid Search רגיל, אלא רק על חלקם.

קוד נמצא כאן

פרק 2 – ML-Based Malware Detection (זיהוי תוכנות זדוניות ע"י ML)

קודים של הפרק: https://github.com/PacktPublishing/Machine-Learning-for-Cybersecurity-Cookbook/tree/master/Chapter02



ניתוח סטטי (Malware static analysis)

בניתוח סטטי אנחנו בודקים מדגם מבלי לבצע אותו (סריקת תוכנה מבלי להפעיל אותה). ניתן להשיג בניתוח סטטי כמויות גדולות של מידע כמו שם הקובץ ועד חתימות YARA מיוחדות. עם זאת, ניתוח סטטי הוא עדיין לא אמין במיוחד (כיוון שניתן לטשטש פרטים חיצוניים בתוכנה), ותחליף טוב הוא ניתוח דינמי (בהמשך).

בחלק הראשון של הפרק, כתבנו קוד שמוצא את ה־MD5 וה־SHA256 של תוכנה מסוימת.

בחלק השני למדנו מהי YARA (שפת מחשב שמאפשרת למומחה אבטחה להגדיר תנאים ולהריץ אותם על תוכנה, ובכך לראות האם התוכנה מממשת את החוקים), והרצנו חוקי YARA על קבצים.

בחלק השלישי ניתחנו header של קבצי PE.

קודים: כאן, כאן, כאן וכאן

ניתוח דינמי (Malware dynamic analysis)

ניתוח דינמי הוא בדיקת מדגם עם הפעלתו, בשונה מניתוח סטטי. עשינו זאת באמצעות Cuckoo sandbox.

שימוש ב־ML לזיהוי סוג קובץ (Detect file type)

בשלב הראשון, ניצור סט מידע. השתמשנו לצורך כך ב־API של GitHub. בשלב השני יצרנו מודל Random Forest.

קודים נמצאים כאן וכאן

מדידת דמיון בין שתי מחרוזות (Measuring similarity)

כדי להשוות בין שתי מחרוזות אפשר להשוות אותם בדרך הכי פשוטה, ותוחזר לנו תוצאת כן/לא. אך לפעמים נרצה תוצאה מספרית שמודדת דמיון בין שתי מחרוזות, במקום משווה אותם באופן הגס. לצורך כך נשתמש ב־ssdeep.

קוד נמצא כאן

מדידת דמיון בין שני קבצים

גם כאן, אפשר לאמוד דמיון בין שני קבצים בינאריים/טקסטואליים באמצעות ssdeep.

קוד נמצא בעמוד 69

חילוץ N-grams

N-grams הם רצף כאשר N מייצג מספר טוקנים (ראה: "עיבוד שפה טבעית"). אם יש את המשפט "בוקר טוב לכולם, היום יום רביעי", אז:

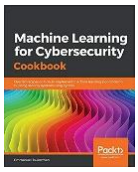
N=1: בוקר, טוב, לכולם, היום, יום, רביעי.
N=2: בוקר טוב, טוב לכולם, לכולם היום, היום יום, יום רביעי.
N=3: בוקר טוב לכולם, טוב לכולם היום, לכולם היום יום, היום יום רביעי.

חילוץ N-grams של תוכנות מאפשר לראות את הבתים שמאפשרים לנו לזהות תוכנה זדונית (בערך).

קוד נמצא כאן

בחירת ה־N-grams הטובים ביותר

במתכון זה בוחרים את ה־N-grams האופטימליים.



קוד נמצא כאן

בניית גלאי תוכנה זדונית סטטי (Static malware detector)

במתכון הזה מחברים את כל הידע של המתכונים הקודמים לבניית גלאי/מזהה תוכנה זדונית בשיטה הסטטית (ללא פתיחת התוכנה).

יש מתכונים נוספים בפרק זה...

פרק 3 – Advanced Malware Detection (זיהוי מתקדם של תוכנה זדונית)

זיהוי JavaScript מוסתר

לפעמים יש קוד JS מוסתר בתוך קובץ (לדוגמה PDF), והוא עושה פעולה (אולי זדונית) כלשהי ללא ידיעתנו. במתכון זה לומדים כיצד לזהות את הקוד והאם יש כזה.

ייצוא מאפיינים מקובץ PDF

לפעמים נרצה לנתח קבצי PDF כדי להשתמש בהם לכל מיני דברים. בהידרים של הקובץ נוכל לגלות כל מיני דברים ובניהם האם הקובץ זדוני (לדוגמה בהידר JavaScript=1, נסיק שיש ב־PDF קוד JavaScript, בין אם הוא מוסתר או לא, ונדע שיש איזה משהו שפועל מאחורי הקלעים ללא ידיעתנו).

פרק 4 – ML for Social Engineering (ML עבור הנדסה חברתית)

בוט פישינג ל־Twitter

הבוט ישתמש בבינה מלאכותית כדי לחקות את הטוויטים היעודיים. באותם טוויטים יהיו לינקים, שיהיו קישורי פישינג.

התחזות קולית