Krytyczne wykorzystanie AI/ML w cyberbezpieczeństwie, w tym generatywna sztuczna inteligencja

Małgorzata Plichta, Damian Kostycz, Paweł Murdzek, Piotr Szewczyk
Instytut Telekomunikacji, Politechnika Warszawska

***Abstract*—W obliczu dynamicznie zmieniających się standardów oraz stale rozwijających się technologii, często brakuje spójnych i skutecznych metod umożliwiających sukcesywne zarządzanie cyberbezpieczeństwem. Jednym z głównych wyzwań stawianych obecnie specjalistom w tej dziedzinie jest integracja złożonych systemów informatycznych z narzędziami i frameworkami z zakresu bezpieczeństwa. W niniejszym artykule przedstawiono wyniki przeglądu literatury oraz eksperymentów przeprowadzonych z wykorzystaniem metod sztucznej inteligencji i uczenia maszynowego (AI/ML) na dwóch zbiorach danych dotyczących wykrywania zagrożeń sieciowych oraz ataków typu APT. Szczególną uwagę poświęcono detekcji podejrzanych zachowań użytkowników systemu Windows, analizie anomalii w ruchu sieciowym, etapom przetwarzania danych (preprocessingu), a także zastosowaniu technik generatywnej sztucznej inteligencji.**

.

***Index Terms*—cyberbezpieczeństwo, AI, ML, IDS, APT, generatywna AI, anomaly detection, cyberattacks, DDoS, Windows**

I. Wprowadzenie

Współcześnie, w dobie rozwoju sztucznej inteligencji oraz wszechobecnej automatyzacji, nie sposób ignorować rosnących kompetencji cyberprzestępców w wykorzystywaniu AI do własnych celów [1]. Algorytmy uczenia maszynowego są coraz częściej stosowane nie tylko w działaniach obronnych, ale także ofensywnych – od automatyzacji phishingu, przez tworzenie przekonujących deepfake'ów, po przeprowadzanie ataków typu APT (Advanced Persistent Threat) przy wsparciu modeli generatywnych [2], [3].

Z drugiej strony, rosnące zapotrzebowanie na rozwiązania klasy AI-based security sprawia, że firmy i instytucje badawcze prześcigają się w opracowywaniu nowych technik ochrony infrastruktury IT. Szczególne miejsce zajmują tu rozwiązania klasyfikacyjne oparte na analizie anomalii w ruchu sieciowym, systemach wykrywania intruzów (IDS), czy adaptacyjne firewalle wspierane przez algorytmy uczenia głębokiego [4], [5]. Przykładem takich narzędzi mogą być nowoczesne systemy klasy EDR (Endpoint Detection and Response), które wykorzystują sztuczną inteligencję do identyfikacji i neutralizacji zagrożeń w czasie rzeczywistym [6]. Choć niektóre z tych rozwiązań mają charakter komercyjny i ograniczony dostęp do szczegółów implementacyjnych, ich znaczenie rośnie z każdym rokiem.

Jednocześnie nie można pominąć zagrożeń związanych z bezpieczeństwem samych modeli AI/ML – w tym ataków typu adversarial, model stealing, data poisoning czy inference attacks [7]–[9]. Pojawiają się zatem dwa równoległe kierunki badań: wykorzystanie AI do ochrony oraz ochrona AI jako technologii strategicznej.

W niniejszym artykule przedstawiono przegląd literatury oraz wyniki badań eksperymentalnych z wykorzystaniem technik AI/ML w obszarze wykrywania zagrożeń. Główne działania dotyczyły analizy danych i konstrukcji modeli ML, których celem było wskazanie wydajnych metod klasyfikacji podejrzanych zachowań użytkowników systemu Windows oraz anomalii w ruchu sieciowym. Dodatkowo, omówiono możliwości zastosowania generatywnej sztucznej inteligencji zarówno jako zagrożenia, jak i potencjalnego narzędzia wspierającego analizę incydentów bezpieczeństwa.

II. Przegląd literatury

Współczesna dyskusja naukowa na temat automatyzacji i wykrywania ataków często opiera się na poszukiwaniu najlepszych i najwydajniejszych metod oraz ich dopasowaniu do poszczególnych typów zagrożeń [10]. W wielu publikacjach badacze podkreślają rosnące znaczenie aspektów cloud computing oraz konieczność badania bezpieczeństwa środowisk chmurowych. W przeglądzie „A comprehensive review of AI based intrusion detection system" autorzy skupiają się na podziale systemów detekcji (Intrusion Detection System) na te bazujące na sieci, hostach oraz hybrydowe (network-based, host-based, hypervisor-based, distributed-based). Podobne obserwacje przedstawia również analiza opublikowana przez GeeksforGeeks [4], gdzie zaakcentowano rolę sztucznej inteligencji w zwiększaniu skuteczności współczesnych systemów bezpieczeństwa – w szczególności dzięki zdolności AI do przetwarzania dużych zbiorów danych w czasie rzeczywistym, wykrywania anomalii behawioralnych oraz automatyzacji reakcji na incydenty. Artykuł ten ukazuje także rosnące znaczenie predykcyjnego charakteru narzędzi AI w przeciwdziałaniu zagrożeniom takim jak ransomware, phishing czy malware, co wzmacnia konieczność dalszych badań nad ich skutecznością i adaptacyjnością w środowiskach wielochmurowych.

Poruszona tematyka jest istotna, gdyż dyskusja przekierowuje uwagę specjalistów ds. bezpieczeństwa na inny wektor ochrony, przy jednoczesnym wskazaniu ograniczeń frameworków zajmujących się wykrywaniem anomalii (anomaly detection) oraz bazowania na tradycyjnych

firewallach w sieciach opartych na rozwiązaniach wielochmurowych [10].

Kolejny istotny wątek poruszony w publikacji badaczy z indyjskiego Christ University to zastosowanie technik sztucznej inteligencji – w szczególności uczenia maszynowego (ML), głębokiego uczenia (DL) oraz metod zespołowych (ensemble learning) – do wykrywania włamań i klasyfikacji ataków. Autorzy wskazują, że metody oparte na AI oferują większą dokładność i skuteczność w porównaniu do tradycyjnych podejść, jednak większość badań skupia się na ogólnej detekcji ataków, pomijając aspekt dokładnego rozróżnienia poszczególnych typów zagrożeń. Zwraca się również uwagę na wyzwania związane z wieloklasową klasyfikacją ataków oraz potrzebę dalszego rozwoju metod umożliwiających automatyczne uczenie się cech z dużych zbiorów danych, co jest szczególnie istotne w kontekście ciągle rosnącej ilości danych w chmurze.

W podobnym kontekście praktyczne zastosowania AI w systemach detekcji zagrożeń przedstawia zestawienie opublikowane przez Analytics Insight, gdzie opisano wybrane rozwiązania komercyjne, takie jak Darktrace czy CrowdStrike Falcon. Systemy te wykorzystują algorytmy uczenia maszynowego i analizy behawioralnej do identyfikacji anomalii i reagowania na ataki w czasie rzeczywistym, również w środowiskach wielochmurowych. Pokazuje to, że kierunek wyznaczony przez badania akademickie znajduje już swoje odzwierciedlenie w implementacjach przemysłowych, co wzmacnia znaczenie dalszych badań nad efektywnością i precyzją AI w kontekście bezpieczeństwa sieciowego [11].

Zbieżne wnioski przedstawia również Vidura Wijekoon [12], wskazując, że AI znacząco zwiększa skuteczność systemów IDS – m.in. poprzez redukcję fałszywych alarmów i lepsze wykrywanie ataków typu zero-day. Autor podkreśla także rolę algorytmów takich jak Random Forest czy SVM w identyfikacji anomalii, co potwierdza skuteczność podejść opartych na uczeniu maszynowym w dynamicznych środowiskach sieciowych i chmurowych.

Analiza opublikowana na platformie GeeksforGeeks [4] z kolei wskazuje, że zastosowanie sztucznej inteligencji w cyberbezpieczeństwie umożliwia analizę ogromnych ilości danych w czasie rzeczywistym, co pozwala na skuteczniejsze wykrywanie złośliwego oprogramowania, ataków typu ransomware oraz phishingu. Podkreślono również znaczenie automatyzacji reakcji na incydenty oraz przewidywania potencjalnych zagrożeń na podstawie wzorców behawioralnych. Wskazuje to na coraz większą dojrzałość i praktyczne znaczenie narzędzi AI nie tylko w środowiskach akademickich, ale i w realnych zastosowaniach systemowych.

III. Implementacja rozwiazania na zbiorze danych CIC-IDS2017

W niniejszym rozdziale opisano szczegóły implementacji systemu wykrywającego zagrożenia w ruchu sieciowym z wykorzystaniem metod uczenia maszynowego. System ten obejmuje cztery główne komponenty: moduł przygotowania danych, moduł treningowy, warstwę ewaluacyjną oraz moduł odpowiedzialny za analizę ruchu w czasie rzeczywistym.

A. Zbiór danych

Zbiór danych CIC-IDS2017 (Canadian Institute for Cybersecurity – Intrusion Detection System 2017) stanowi kompleksowy zestaw danych służący do badania problematyki wykrywania intruzji w sieciach komputerowych [13]. Obejmuje on zarówno ruch normalny (benign), jak i dane dotyczące różnorodnych ataków, takich jak brute force, DoS/DDoS, ataki webowe (m.in. XSS, SQL injection), skanowanie portów, a także działania zaawansowanych zagrożeń typu APT. Dane zostały zebrane w kontrolowanym środowisku laboratoryjnym w ciągu pięciu dni i zawierają metadane przepływów sieciowych (flow-based features) generowane przy użyciu narzędzi takich jak CICFlowMeter.

Dzięki etykietowaniu próbek oraz zachowaniu zróżnicowanych scenariuszy ataków, zbiór umożliwia zarówno zastosowanie metod nadzorowanych, jak i nienadzorowanych w analizie anomalii. CIC-IDS2017 jest uznawany za jedno z bardziej reprezentatywnych i realistycznych źródeł danych do testowania algorytmów detekcji zagrożeń w systemach IDS.

B. Wstępne przetwarzanie danych

Pierwszym krokiem w implementacji systemu było przygotowanie danych wejściowych do trenowania modeli uczenia maszynowego. Dane poddano czyszczeniu poprzez usunięcie nieistotnych kolumn oraz rekordów z brakującymi wartościami. Następnie dokonano kodowania zmiennych kategorycznych (takich jak protokół sieciowy) do postaci liczbowej, co umożliwiło ich wykorzystanie w modelach klasyfikacyjnych.

W celu zapewnienia jednolitej skali wartości, wszystkie cechy zostały przeskalowane przy użyciu standaryzacji. Kluczowym etapem było także zbalansowanie zbioru danych treningowych przy pomocy techniki SMOTE, co ograniczyło wpływ dominujących klas na wynik modelu. SMOTE (Synthetic Minority Over-sampling Technique) polega na sztucznym dodawaniu nowych przykładów do mniej licznych klas, tak aby zbiór danych zawierał podobną liczbę próbek dla każdej kategorii. Nowe dane tworzone są poprzez interpolację — czyli generowanie przykładów na podstawie istniejących próbek klasy mniejszościowej i ich najbliższych sąsiadów w przestrzeni cech.

Na koniec dane zostały podzielone na zestawy treningowe i walidacyjne, a osobno przygotowano dane testowe, które nie były używane podczas trenowania modelu.

C. Trenowanie modeli detekcyjnych

W celu wykrywania anomalii i potencjalnych zagrożeń w analizowanym ruchu sieciowym zastosowano dwa odrębne podejścia modelowania: las losowy (*Random Forest*) oraz głęboką sieć neuronową (*Deep Neural Network*, DNN). Oba modele zostały dobrane ze względu na ich udokumentowaną skuteczność w zadaniach klasyfikacyjnych, ale reprezentują różne klasy algorytmów: metody zespołowe oraz głębokie uczenie.

Random Forest to algorytm klasyfikacyjny wykorzystujący zbiór niezależnych drzew decyzyjnych, którego końcowa decyzja opiera się na mechanizmie głosowania większościowego.

W niniejszej pracy skonstruowano las losowy składający się ze 100 drzew, z których każde trenowane było na losowej próbce danych wejściowych (z dobieraniem ze zwracaniem, czyli tzw. próbkowaniem replikowanym). Dodatkowo, w celu redukcji korelacji pomiędzy drzewami, w każdym węźle decyzyjnym uwzględniano losowy podzbiór cech. Model ten wykazuje wysoką odporność na przeuczenie, co czyni go odpowiednim wyborem dla zbiorów danych zawierających szum lub cechy o niewielkiej istotności. Dodatkową zaletą jest możliwość interpretacji wyników poprzez analizę ważności poszczególnych cech wejściowych. Random Forest nie wymaga wcześniejszego skalowania danych ani ich liniowego rozdzielania, co czyni go wszechstronnym narzędziem w klasyfikacji tablicowych danych wejściowych [14].

Drugim zastosowanym podejściem była głęboka sieć neuronowa (DNN) zbudowana z kilku warstw w pełni połączonych (*fully connected*, dense). Sieć składała się z warstwy wejściowej dostosowanej do 42 cech wejściowych, dwóch warstw ukrytych zawierających odpowiednio 128 i 64 neurony z funkcją aktywacji ReLU oraz warstwy wyjściowej z pojedynczym neuronem i funkcją aktywacji sigmoidalnej (dla klasyfikacji binarnej). Funkcja ReLU (*Rectified Linear Unit*) wprowadza nieliniowość, przepuszczając dodatnie wartości bez zmian, a ucinając wartości ujemne do zera, co sprzyja szybszemu uczeniu głębokich modeli. Funkcja sigmoidalna natomiast przekształca wynik wyjściowy do przedziału od 0 do 1, dzięki czemu może być interpretowana jako prawdopodobieństwo przynależności do klasy pozytywnej. W celu ograniczenia przeuczenia zastosowano mechanizm *Dropout* (z prawdopodobieństwem 0,3–0,5), który losowo wyłącza część neuronów w czasie uczenia, co zmusza model do uogólniania. Uczenie odbywało się z użyciem algorytmu optymalizacji *Adam*, który łączy zalety metod adaptacyjnych i momentu, automatycznie dostosowując tempo uczenia dla każdej wagi w modelu. Zastosowano funkcję straty *binary_crossentropy*, mierzącą różnicę między przewidywanym prawdopodobieństwem a rzeczywistą etykietą klasy w zadaniach binarnych. Ponadto wykorzystano technikę *Early Stopping*, która przerywa trenowanie, gdy jakość modelu na zbiorze walidacyjnym przestaje się poprawiać, zapobiegając jego przeuczeniu [15].

Zaletą zastosowanej sieci neuronowej jest jej zdolność do modelowania złożonych, nieliniowych zależności w danych, co może prowadzić do wyższej skuteczności klasyfikacyjnej. Jednakże, z uwagi na większe wymagania obliczeniowe oraz trudności interpretacyjne, model ten wymaga ostrożnej kalibracji hiperparametrów i weryfikacji skuteczności działania.

D. Ewaluacja i analiza wyników

Ewaluacja skuteczności modeli została przeprowadzona zarówno na zbiorze walidacyjnym, jak i testowym. Dla każdego modelu wyliczono standardowe metryki klasyfikacji: dokładność (accuracy – ogólny odsetek poprawnych predykcji), precyzję (precision – jak wiele zgłoszonych alarmów było rzeczywistymi zagrożeniami, czyli niska liczba fałszywych alarmów), czułość (recall – jak wiele rzeczywistych zagrożeń zostało wykrytych, czyli niska liczba pominiętych ataków) oraz miarę F1 (F1-score – kompromis między precyzją a czułością). Ponadto, dla sieci neuronowej wygenerowano wykresy pokazujące przebieg procesu uczenia (fig. 1) – dokładność oraz funkcję straty modelu.

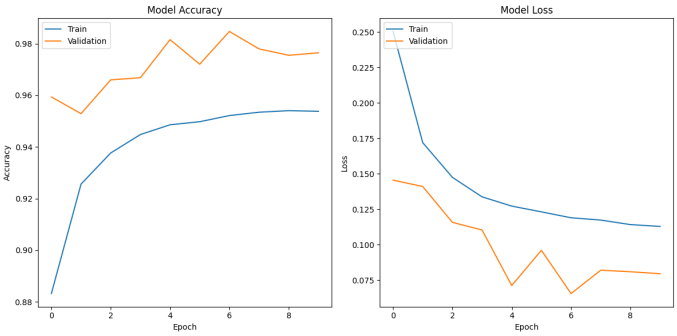

Fig. 1: Proces uczenia się sieci neuronowej

Poniżej zaprezentowano wyniki uzyskane przez oba modele w trybie binarnej klasyfikacji zagrożeń.

TABLE I: Wyniki na zbiorze walidacyjnym (klasyfikacja binarna)

Model	Accuracy	Precision	Recall	F1-score
Random Forest	0.98	0.98	0.98	0.98
Sieć neuronowa	0.45	0.20	0.45	0.28

TABLE II: Wyniki na zbiorze testowym (klasyfikacja binarna)

Model	Accuracy	Precision	Recall	F1-score
Random Forest	0.90	0.91	0.90	0.90
Sieć neuronowa	0.32	0.10	0.32	0.15

Rezultaty wskazują jednoznacznie na wyższą skuteczność klasyfikatora Random Forest względem sieci neuronowej. Model drzew decyzyjnych osiągał nie tylko wysoką dokładność, ale również dobrą równowagę pomiędzy precyzją a czułością. W przypadku sieci neuronowej zauważono znaczne problemy z klasyfikacją klasy ataku, co może wskazywać na konieczność lepszego dostrojenia hiperparametrów oraz wydłużenia treningu.

Stworzono również macierz pomyłek zarówno dla sieci neuronowej, jak i dla modelu Random Forest, aby lepiej zrozumieć decyzje podejmowane przez modele podczas ich działania. Obie macierze zostały stworzone na podstawie pracy modeli nad testowym zbiorem danych. Na fig. 2 można zauważyć, że model kompletnie nie wykrywa zagrożeń, wszystkie testowe dane wejściowe, które reprezentowały złośliwy ruch, zostały zaklasyfikowane jako ruch niestanowiący zagrożenia. Świadczy to o dużym problemie w zbiorze danych lub w nieodpowiednim dostrojeniu treningu modelu przy pomocy hiperparametrów.

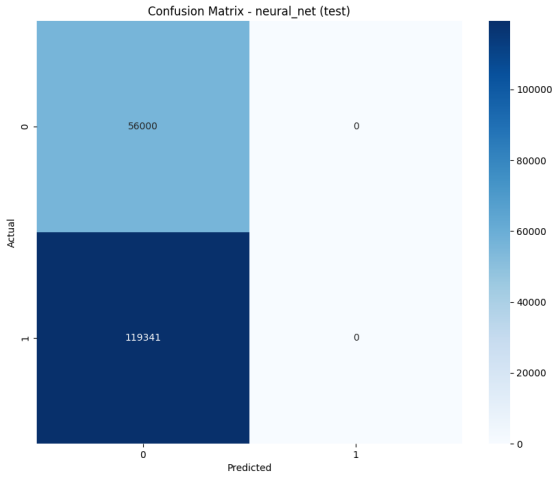

Fig. 2: Macierz pomyłek modelu opartego na sieci neuronowej

Natomiast na fig. 3 przedstawiono macierz pomyłek dla modelu Random Forest. Widać na niej, że predykcje wykonywane przez model w znakomitej większości były prawdziwe. Jak przedstawiono w tabeli III, model Random Forest osiąga wysokie wartości metryk (ok. 90%) i mógłby zostać użyty np. do wczesnego alarmowania o podejrzanym ruchu w sieci.

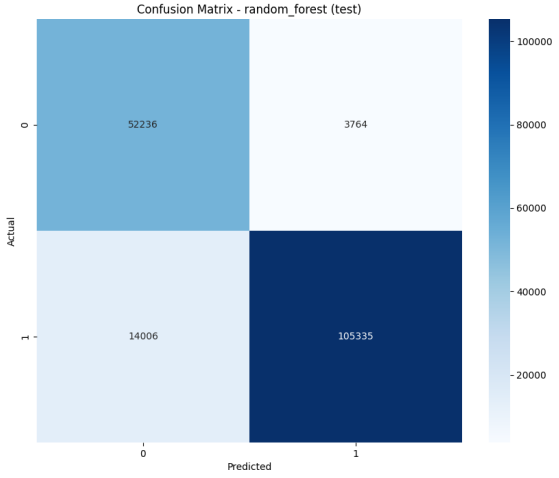

Fig. 3: Macierz pomyłek dla modelu Random Forest

E. Detekcja zagrożeń w czasie rzeczywistym

Dla zapewnienia praktycznego zastosowania systemu opracowano również moduł monitorujący działający w czasie rzeczywistym. Komponent ten analizował ruch sieciowy na zadanym interfejsie, wybierał odpowiednie cechy z pakietów, przetwarzał je zgodnie ze sposobem treningowym, a następnie klasyfikował jako zagrożenie lub normalny ruch.

Każdy wykryty incydent zapisywano do pliku w postaci raportu zawierającego adresy IP, protokół, rozmiar pakietu, etykietę klasyfikacyjną oraz wartość pewności predykcji. Zebrane dane były wykorzystywane do tworzenia wizualizacji zagrożeń, w tym rozkładów zagrożeń według protokołu oraz wykresów aktywności w czasie.

Dzięki zastosowaniu modeli uprzednio zapisanych po treningu, detekcja w czasie rzeczywistym nie wymagała ponownego uczenia i mogła być uruchamiana wielokrotnie w środowiskach testowych. Działanie tego modułu przedstawiono na fig. 4.

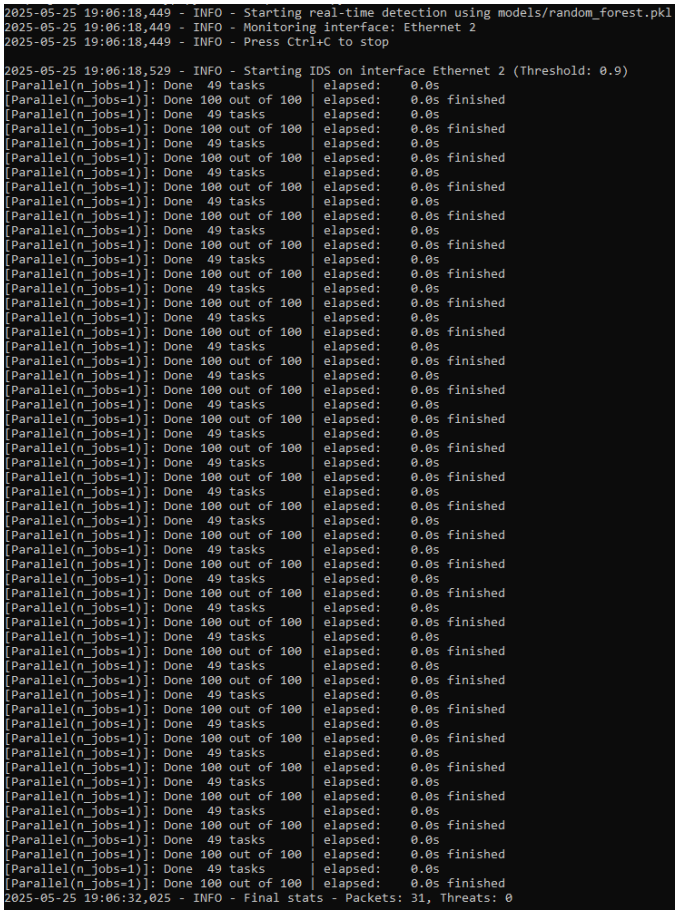

Fig. 4: Działanie modułu sprawdzającego zachowanie modelu w czasie rzeczywistym

F. Podsumowanie

Zaprojektowany system stanowi kompletne rozwiązanie do analizy i detekcji zagrożeń w sieciach komputerowych. Dzięki modularnej strukturze możliwe było szybkie eksperymentowanie z różnymi podejściami oraz skalowalne wdrożenie zarówno offline (trening i ewaluacja), jak i online (monitoring w czasie rzeczywistym). Zastosowanie zbioru danych CIC-IDS2017 oraz replikowalnych pipeline'ów pozwala na wiarygodną ocenę skuteczności modeli i dalszy rozwój w kierunku rzeczywistych aplikacji w środowiskach produkcyjnych.

IV. Eksperyment na datasecie *Unraveled: A Semi-Synthetic Dataset for APT*

A. Wprowadzenie

Zbiór danych Unraveled, a w szczególności jego część znajdująca się w katalogu data/host-logs/windows, został stworzony z myślą o analizie aktywności na poziomie hosta w

kontekście zaawansowanych, trwałych zagrożeń (APT – Advanced Persistent Threats) [16]. Dane te obejmują szczegółowe logi systemowe z systemów Windows, zawierające informacje o uruchamianych procesach, poleceniach oraz zdarzeniach uwierzytelniania. Dzięki odpowiednio oznakowanym rekordom, umożliwiają one badanie wzorców zachowań użytkowników oraz korelowanie ich z działaniami atakujących o różnym poziomie zaawansowania, w tym grup APT. Logi te stanowią materiał badawczy dla oceny metod detekcji anomalii oraz modeli uczenia maszynowego do wczesnego wykrywania zagrożeń wewnętrznych i zewnętrznych w środowiskach korporacyjnych (systemy typu IDS).

Pierwszym krokiem była normalizacja datasetów i zagregowanie logów z różnych stacji Windows.

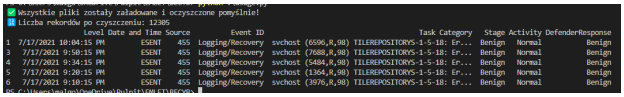

Fig. 5: Dataset po normalizacji

Po wczytaniu i oczyszczeniu danych, skrypt przechodzi do etapu inżynierii cech oraz trenowania czterech różnych modeli detekcji anomalii: Isolation Forest, MLPClassifier, Autoencoder oraz klasyfikatora binarnego w architekturze Keras. Dane zostają przetworzone za pomocą transformacji numerycznych i kategorycznych, co umożliwia ich efektywne wykorzystanie w modelach uczenia maszynowego.

Zestaw danych wejściowych obejmował ponad pół miliona rekordów, które po przefiltrowaniu i oczyszczeniu ograniczono do ok. 8600 przykładów. Pomimo znacznie niezbalansowanego zbioru danych (jedynie pojedyncze anomalie), modele uzyskały bardzo wysoką dokładność — szczególnie MLP i klasyfikator binarny, które osiągnęły 100 procent skuteczności na zbiorze testowym.

Autoencoder z kolei wykazał się zdolnością wykrywania anomalii przy użyciu błędu rekonstrukcji (MSE), skutecznie identyfikując przypadek odstający mimo bardzo niskiej precyzji. Wyniki te sugerują, że klasyczne klasyfikatory dobrze dopasowują się do zrównoważonych danych, natomiast modele nienadzorowane, jak Isolation Forest i Autoencoder, są bardziej odporne na brak etykiet, ale ich skuteczność może być ograniczona przez niską reprezentację klas anomalii.

Zebrane dane po wstępnym oczyszczeniu zawierały 8621 rekordów, z których jedynie 1 przypadek został oznaczony jako anomalia, co znacząco wpłynęło na ocenę skuteczności poszczególnych modeli. Dla przykładu, klasyfikator MLP uzyskał perfekcyjny wynik — 100 procent precyzji, recallu i f1-score — jednak należy zauważyć, że przy tak ekstremalnie niezrównoważonym zbiorze (1:1724), nawet minimalna liczba błędów może drastycznie zaniżyć ocenę innych modeli.

TABLE III: Wyniki na zbiorze testowym (klasyfikacja binarna)

Model	Accuracy	Precision	Recall	F1-score
Isolation Forest	0.88	0.00	0.00	0.00
MLP	1.00	1.00	1.00	1.00
Autoencoder	0.75	0.00	0.00	0.00
Keras Classifier	0.88	0.00	0.00	0.00

Isolation Forest oraz Autoencoder osiągnęły wysoką skuteczność w wykrywaniu przypadków normalnych (96–95 procent recall), ale miały trudności z poprawnym rozpoznaniem pojedynczej anomalii (f1-score dla klasy 1 wynosiło 0.0 i 0.02 odpowiednio). Warto również zauważyć, że Autoencoder, mimo bardzo niskiego błędu rekonstrukcji (MSE 0.0003), klasyfikował anomalię poprawnie — co wskazuje, że może on być bardziej czuły na subtelne odchylenia, nawet przy braku wyraźnej reprezentacji klasy rzadkiej w danych treningowych.

W dalszym ciągu nastąpiło zwiększenie progu wykrywania anomalii (szczególnie w Isolation Forest, gdzie można dobierać parametry i szybko widzieć bezpośredni wpływ na wyniki) oraz konieczne okazało się radykalniejsze zbalansowanie danych, lecz działania te nie przyniosły wymiernych efektów. W naszym odczuciu zbiór danych wymaga strukturalnego ujednolicenia i lepszego odróżnienia danych normalnych od anomalii, ponieważ przy tak niewielkich różnicach kalibracja progów detekcji (szczególnie dla metod opartych na rekonstrukcji) jest obarczona wysokim błędem pomiarowym.

Podobnie jak Isolation Forest, Autoencoder nie wykrył anomalii — f1-score dla klasy 1 wyniósł 0.00, a MSE rekonstrukcji (0.0796) sugeruje, że błędy nie były wystarczająco wysokie, by przekroczyć próg decyzyjny. Wyodrębniony na etapie przetwarzania danych zbiór testowy i walidacyjny był jednak zbyt mały, by umożliwić miarodajną ocenę skuteczności modelu. W efekcie, metryki takie jak accuracy mogą być mylące, a pozostałe wyniki — niestabilne i silnie zależne od konkretnego podziału danych. Wyniki należy więc traktować jako eksperymentalne i wymagające weryfikacji na większym, bardziej zrównoważonym zbiorze testowym.

V. Wnioski i dalsze kierunki

W obliczu rosnącej złożoności cyberataków oraz coraz częstszego wykorzystywania nowych technologii przez cyberprzestępców, rozwój skutecznych metod obrony staje się priorytetem. Sztuczna inteligencja (AI) oraz uczenie maszynowe (ML) oferują ogromny potencjał w tej dziedzinie, jednak ich wdrożenie wiąże się z szeregiem wyzwań.

A. Przygotowanie danych treningowych

Jednym z kluczowych problemów w stosowaniu AI do wykrywania zagrożeń jest niezrównoważony charakter danych. Typowe zbiory danych są zdominowane przez normalny ruch sieciowy, a podejrzane aktywności stanowią ich niewielki odsetek. W celu przeciwdziałania temu zjawisku stosuje się m.in. technikę SMOTE, polegającą na generowaniu dodatkowych przykładów rzadkich zdarzeń. Alternatywą jest metoda ADASYN, która koncentruje się na tworzeniu danych dla trudniejszych do sklasyfikowania przypadków. Coraz większe zainteresowanie budzą również generatywne sieci przeciwstawne (GAN), umożliwiające tworzenie realistycznych scenariuszy ataków na potrzeby trenowania modeli AI.

B. Wydajność i jakość algorytmów

Głębokie uczenie (DL), jako zaawansowana forma AI, wymaga znacznych zasobów obliczeniowych. Aby us-

prawnić działanie systemów wykrywających zagrożenia, opracowywane są wyspecjalizowane architektury, takie jak sieci LSTM (analiza danych czasowych) czy grafowe sieci konwolucyjne (GCN), które badają relacje między urządzeniami w sieci. Technika transfer learning, polegająca na przenoszeniu wiedzy z jednego obszaru do innego, pozwala wykorzystać istniejące modele (np. z obszaru rozpoznawania obrazów) w kontekście cyberbezpieczeństwa.

C. Integracja z narzędziami bezpieczeństwa

Skuteczność AI zależy nie tylko od jakości algorytmów, lecz także od możliwości ich integracji z istniejącymi systemami ochrony. Platformy takie jak SOAR (Security Orchestration, Automation and Response) pozwalają na automatyzację reakcji na incydenty, np. blokowanie niebezpiecznych adresów IP. Modele AI mogą również współdziałać w ramach systemów hybrydowych, łączących proste algorytmy (np. Random Forest) z zaawansowanymi modelami (np. autoenkodery), umożliwiającymi wykrywanie nieznanych dotąd zagrożeń.

D. Generatywna AI: szanse i zagrożenia

Generatywne modele AI, zdolne do tworzenia tekstu, kodu czy obrazów, otwierają nowe możliwości, ale też stwarzają nowe ryzyka. Narzędzia takie jak ChatGPT mogą wspomagać analityków w interpretacji danych logowania, lecz jednocześnie mogą być wykorzystywane przez przestępców do tworzenia wiarygodnych wiadomości phishingowych. Badacze eksperymentują z wykorzystaniem GAN do symulowania ataków, co pozwala lepiej przygotować systemy obronne na rzeczywiste zagrożenia.

E. Bezpieczeństwo i aspekty etyczne

Zastosowanie AI w cyberbezpieczeństwie pociąga za sobą konieczność ochrony samych modeli przed atakami. Przykładem są ataki typu adversarial, polegające na celowej modyfikacji danych wejściowych w celu zmylenia systemu. Istnieje też ryzyko kradzieży modeli, które mogłyby zostać wykorzystane przeciwko ich twórcom. Rozwiązaniem są techniki szyfrowania modeli oraz rozwój wyjaśnialnej AI (XAI), która pozwala lepiej zrozumieć decyzje podejmowane przez sieci neuronowe. Równie istotne są kwestie etyczne, m.in. dotyczące prywatności pracowników oraz odpowiedzialności za automatyczne decyzje podejmowane przez systemy.

F. Weryfikacja w praktycznych zastosowaniach

Chociaż wiele modeli AI wykazuje wysoką skuteczność w środowiskach testowych, ich efektywność w rzeczywistych systemach informatycznych bywa ograniczona. Kluczowe znaczenie ma testowanie rozwiązań w realnych warunkach korporacyjnych oraz współpraca z instytucjami takimi jak MITRE, która rozwija framework ATT&CK. Pozwala on klasyfikować taktyki i techniki ataków, wspierając adaptację modeli AI do aktualnych zagrożeń.

G. Wnioski

Rozwój AI w cyberbezpieczeństwie wymaga współpracy specjalistów z różnych dziedzin. Inżynierowie, eksperci ds. bezpieczeństwa, prawnicy i etycy powinni współdziałać przy tworzeniu systemów, które będą skuteczne, bezpieczne i zgodne z wartościami społecznymi. Tylko zintegrowane podejście zapewni odpowiednie wykorzystanie potencjału sztucznej inteligencji w obronie przed zagrożeniami cyfrowymi.

REFERENCES

[1] C. Advisory, "Artificial intelligence fuels new wave of complex cyber attacks challenging defenders," https://cybersecuritynews.com/artificial-intelligence-in-cyber-attacks/#:~:text=While%20AI-driven%20threat%20detection%20systems%20have%20advanced%2C%20cybercriminals,traditional%20defenses%2C%20creating%20a%20high-stakes%20technological%20arms%20race., 2025, dostęp: 2025-05-20.

[2] T. B. Brown and et al., "Language models are few-shot learners," *NeurIPS*, 2020. [Online]. Available: https://arxiv.org/abs/2005.14165

[3] M. Brundage and et al., "The malicious use of artificial intelligence: Forecasting, prevention, and mitigation," 2018. [Online]. Available: https://arxiv.org/abs/1802.07228

[4] GeeksforGeeks, "Ai in cybersecurity," https://www.geeksforgeeks.org/ai-in-cybersecurity/, 2025, dostęp: 2025-05-20.

[5] Avsistema, "https://avsistema.com/best-ai-powered-firewalls-for-enterprise-security-in-2025/," https://avsistema.com/best-ai-powered-firewalls-for-enterprise-security-in-2025/, 2025, dostęp: 2025-05-20.

[6] M. Corporation, "Endpoint detection and response framework: Mitre attck," 2021. [Online]. Available: https://attack.mitre.org

[7] B. Biggio and et al., "Evasion attacks against machine learning at test time," *ECML PKDD*, 2013.

[8] F. Tramèr and et al., "Stealing machine learning models via prediction apis," *USENIX Security*, 2016.

[9] M. Jagielski and et al., "Manipulating machine learning: Poisoning attacks and countermeasures for regression learning," *IEEE SP*, 2018.

[10] T. Sowmya and E. Mary Anita, "A comprehensive review of ai based intrusion detection system," *Measurement: Sensors*, vol. 28, p. 100827, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2665917423001630

[11] M. Prabhakar, "Top 10 ai-based threat detection systems," 2024, dostęp: 2025-05-19. [Online]. Available: https://www.analyticsinsight.net/artificial-intelligence/top-10-ai-based-threat-detection-systems

[12] V. Wijekoon, "Ai in intrusion detection systems (ids): A game-changer in network security," https://medium.com/@ViduraAI/ai-in-intrusion-detection-systems-ids-a-game-changer-in-network-security-1e514c5753f1, 2024, dostęp: 2025-05-20.

[13] C. H. N, "Network intrusion dataset(cic-ids-2017)," 2023, dostęp: 2025-05-17. [Online]. Available: https://www.kaggle.com/datasets/chethuhn/network-intrusion-dataset?fbclid=IwY2xjawKVV5hleHRuA2FlbQIxMABicmlkETBPcWREVm5QVmplOXpxbnNkAR7UEfNr23-GIfpNVuWUwX_fzCWBmEKzSx2482VMOYOI44Pw5OtpO2JoUjJVZg_aem_KE4WRmGcNJnu8gPbHCe5jg

[14] IBM, "What is random forest?" https://www.ibm.com/think/topics/random-forest, 2025, dostęp: 2025-05-10.

[15] ——, "What is a neural network?" https://www.ibm.com/think/topics/neural-networks, 2021, dostęp: 2025-05-10.

[16] S. Myneni, "Unraveled- a semi-synthetic dataset for advanced persistent threats," https://gitlab.com/asu22/unraveledfbclid=IwY2xjawKVV8xleHRuA2FlbQIxMABicmlkETBPcWREVm5QVmplOXpxbnNkAR5J5vcvvejLZayaM9YXNvXHOatyv0CNIxp6KYiDs0jpVs4uBqJ2Bk2Og0ORJg_aem_F0_cqkXjW3kK3fMrjQw_KQ, 2024, dostęp: 2025-05-17.