Wykorzystanie danych z programu KDD99 do oceny i projektowania systemów IDS

Przemysław Kukiełka, Zbigniew Kotulski

Streszczenie

W ostatnich latach wzrasta znaczenie gałęzi nauki zwanej sztuczną inteligencją. Znajduje ona zastosowanie w wielu dziedzinach jedna z nich są również systemy wykrywania włamań IDS.
Dzięki zdolności generalizacji, czyli wykrywania nie tylko tych ataków, które były prezentowane podczas procesu nauki, ale również wszelkich ataków podobnych do nich oraz nowych typów ataków systemy sztucznej inteligencji są bardzo skuteczną technologia. Mogą się również w sposób dynamiczny dostosowywać do sytuacji w sieci (np. uczyć się nowych zachowań użytkowników czy nowych ataków). Ich zaletą jest to, że nie wymagają budowy skomplikowanych zbiorów reguł i sygnatur odrębnych dla każdej instancji ataków, ponieważ są one tworzone automatycznie w procesie nauki.
Do testowania zastosowań systemów sztucznej inteligencji dla systemów IDS może być wykorzystywany przedstawiony w referacie zbiór danych wejściowych będący produktem programu DARPA (*Defenced Advanced Research Project Agency*), a następnie zmodyfikowany w projekcie KDD (*Knowlegde Discovery and Data Mining Competition*). Dane zawierają symulację normalnego ruchu oraz około 40 typów różnych ataków podzielonych na 4 grupy: Denial of Service, User to Root, Remote to Local oraz Probe. W referacie zostały opisane szczegółowo wyniki, założenia oraz cele obu projektów.
Referat zawiera również analizę zastosowania metod sztucznej inteligencji takich jak: drzewa decyzyjne, sieci neuronowe oparte o zasadę samoorganizacji (SOM), oraz sieci neuronowe MLP(*Multi Layer Perceptron*) w systemach wykrywania włamań.

1. Wstęp

Ilość różnych ataków mogących wystąpić w sieci IP gwałtownie wzrasta wraz ze wzrostem znaczenia Internetu i powstawaniem nowych usług sieciowych. Dlatego bardzo istotna kwestią stają się systemy wykrywania włamań IDS (*Intrusion Detection System*).

Systemy IDS możemy podzielić na dwie grupy: bazujące na sygnaturach oraz bazujące na anomaliach. Pierwszy typ jest wykorzystywany do wykrywania znanych ataków przy wykorzystaniu określonych specyficznych dla nich cech. Na przykład dla ataku SYN Flood taką sygnaturą może to być wystąpienie większej niż 20 liczby pakietów TCP SYN skierowanych do tego samego hosta w ciągu 2 minut, na które nie pojawiła się odpowiedź SYN/ACK. Stworzenie odpowiedniego zestawu sygnatur do wykrywania ataków jest procesem skomplikowanym ponadto agresor, który pozna, jakie cechy ataków były brane pod uwagę podczas tworzenie sygnatury może tak zmodyfikować swój atak, aby ominął system IDS. Drugi rodzaj polega na stworzeniu normalnego profilu działalności użytkownika. Wszelkie odchylenia od tego profilu mogą być uznawane jako próba ataku. System IDS bazujący na anomaliach jest bardziej skuteczny do wykrywania nieznanych, nowych typów ataków. Niestety jego wadą są trudności związane ze zbudowaniem takiego profilu obejmującego normalne zachowania użytkownika chociażby ze względu na to, że zbiór tych zachowań jest znacznie większy od zbioru sygnatur związanych z danym atakiem. System IDS pracujący w oparciu o anomalie może generować znacznie większą liczbę alarmów ze względu na częste zmiany przyzwyczajeń czy zachowań użytkownika. Może to być na przykład próba skorzystania z nowej usługi Internetowej czy zainstalowanie nowego rodzaju oprogramowania korzystającego z sieci IP.

Skutecznym narzędziem służącym wykrywaniu nowych ataków lub zmodyfikowanych wersji dobrze znanych ataków okazały się również systemy bazujące na sztucznej inteligencji. Ich głównymi zaletami są zdolności generalizacji, czyli wykrywania nie tylko tych ataków, które były prezentowane podczas procesu nauki, ale również wszelkich ataków podobnych do nich oraz nowych typów ataków. Dzięki prezentowaniu w procesie nauki również normalnych zachowań użytkownika są wstanie wykrywać wszelkie odchylenia od nauczonych schematów. Dlatego można powiedzieć, że systemy bazujące na sztucznej inteligencji łączą w sobie cechy systemów IDS opartych o sygnatury i o wykrywanie anomalii. Mogą się również w sposób dynamiczny dostosowywać do sytuacji w sieci (np. uczyć się nowych zachowań użytkowników, czy nowych ataków). Ich zaletą jest to, że nie wymagają budowy skomplikowanych zbiorów reguł i sygnatur odrębnych dla każdej instancji ataków, ponieważ są one tworzone automatycznie w procesie nauki.

Do testowania zastosowań systemów sztucznej inteligencji dla systemów IDS może być wykorzystywany przedstawiony dalej zbiór danych wejściowych będący produktem programu DARPA a potem zmodyfikowany w projekcie KDD.

2. Programy oceny systemów IDS

Jedna z istotnych kwestii jest ocena skuteczności działania różnych systemów wykrywania włamań. Głównym parametrem wykorzystywanym do oceny systemów IDS jest dokładność wykrywania każdego typu ataku. Drugim parametrem, który powinien być wykorzystywany do oceny jest ilość fałszywych alarmów generowanych przez system. Zachodzą one w momencie, gdy ruch wynikający z normalnej aktywności użytkownika jest uznawany za atak. Inne czynniki o mniejszym znaczeniu brane pod uwagę to: koszty komercyjnego systemu, łatwość instalacji oprogramowania, wielkość ruchu sieciowego, z jakim IDS może sobie poradzić, ilość wymaganej pamięci operacyjnej oraz wymagania na procesor.

W przeszłości powstało kilka programów, które miały służyć ocenie i porównaniu różnych systemów wykrywania włamań. Zostały one przedstawione w tabeli 1.

autor	liczba porówny-wanych systemów IDS	liczba typów ataków/licz-ba atakowa-nych maszyn	analiza fałszy-wych alarmów	długość normalnego ruchu do pomiaru fałszywych alarmów	użyto zmodyfiko-wanych ataków (stealth)	Komentarze
Puketza 1994	2	4/1	tak	nieznana	nie	automatyczne ataki i ruch związany z telnetem
Debar 1998	3	4/1	tak	nieznana	nie	automatyczne ataki i ruch związany z ftp
Shipley 1999	10	12/4	nie	n/d	tak	porównanie 10 komercyjnych systemów
Durst 1999	4	19/4	tak	godziny	tak	1998 Darpa Real time
Lippman n 2000	10	38/4	tak	tygodnie	tak	1998 DarpaOff-Line

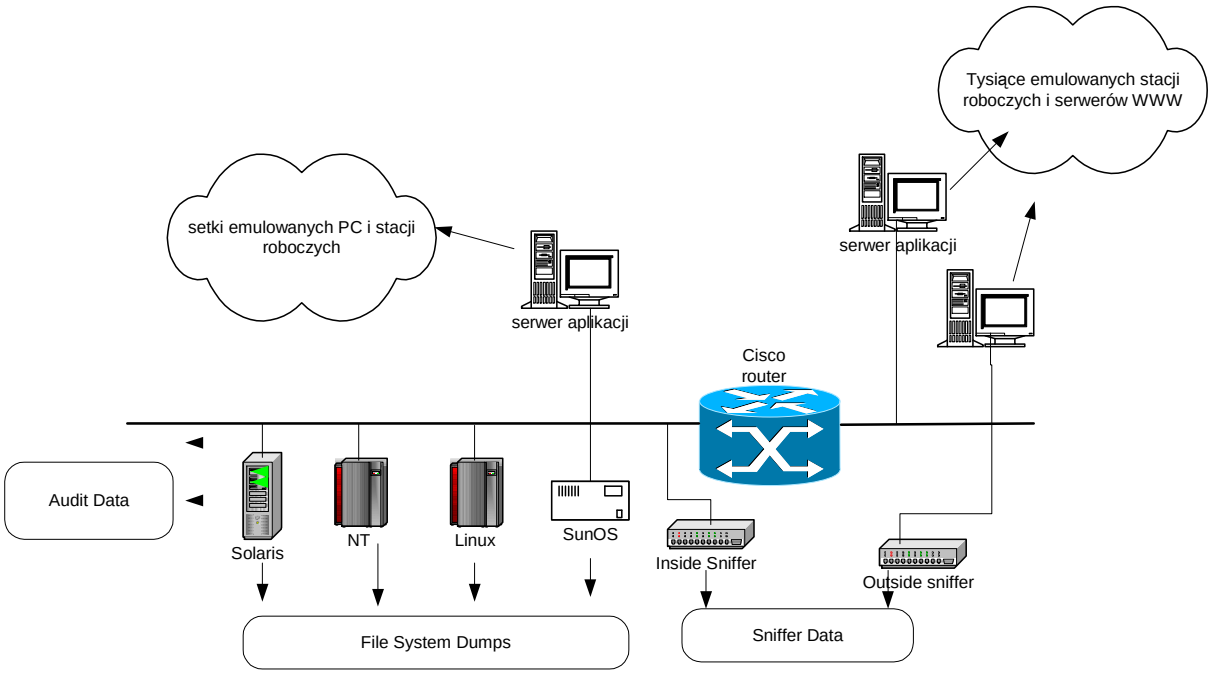
Tabela 1. Charakterystyka programów oceny i porównania systemów IDS [4]

Początkowe systemy oceny brały pod uwagę niewielką liczbę różnych typów ataków, nie rozpatrywano również zmodyfikowanych instancji tego samego ataku. Z czasem ocena stawała się bardziej złożona. W dalszej części pracy zostanie omówiony program DARPA jako najbardziej zaawansowany i często wykorzystywany do badań nad systemami IDS.

3. Program DARPA (Defense Advanced Research Projects Agency)

Każdy uczestnik programu miał za zadanie korzystając z zebranych w sieci testowej danych przetestować swój system IDS i udostępnić rezultaty swoich badań.

Do stworzenia zbioru danych testowych zostały wykorzystane informacje zebrane przez MIT Lincoln Lab w sieci przedstawionej na rysunku 1. Były one zbierane dwukrotnie w roku 1998 i 1999.



Rysunek 1. Architektura sieci użytej do zbierania danych w projekcie DARPA

Dane z wykorzystaniem narzędzia tcpdump w roku 1998 były zbierane tylko na wyjściu WAN routera (*outside sniffer*), a w roku 1999 również na wejściu od strony sieci lokalnej (*inside sniffer*). Dodatkowo, jako tło był symulowany ruch pochodzący od setek emulowanych użytkowników na tysiącach hostów. Symulowane ataki były przeprowadzane przeciw trzem hostom pracującym w oparciu o systemy operacyjne: UNIX, Linux, Sun, a w 1999 dodatkowo również Windows NT. Ponadto zbierane były logi (*audit data*) dla systemu Sun (BSM solaris- wykorzystywane przez niektórych badaczy do wykrywania ataków R2L oraz U2L przeciwko systemom Solaris) oraz NT audit data. Do danych z roku 1998 rok zostały później dodane dodatkowe ataki przeciwko systemom NT oraz zmodyfikowane pod kątem oszukania systemów IDS wersje większości ataków z roku 1998.

4. KDD (Knowledge Discovery and Data Mining Competition)

W roku 1999 w ramach projektu KDD (*Knowledge Discovery and Data Mining Competition*) zebrane w programie DARPA dane zostały przetworzone na poszczególne połączenia tworząc zbiór około 5 milionów rekordów. Połączenie jest rozumiane jako sekwencja pakietów rozpoczynająca się i kończąca w zdefiniowanym czasie, podczas którego dane przepływają pomiędzy adresami źródłowymi i docelowymi przy wykorzystaniu określonego protokołu. Dokładny opis tworzenia tego zestawu danych z wykorzystaniem narzędzi MADAM ID oraz Bro [5] autorzy (W. Lee i S. Stolfo) zamieścili w pracy [1].

Symulowane ataki zostały podzielone na 4 grupy:

- **DoS** (*Denial of Service*) Grupa ataków odmowy usług takich jak np: TCP SYN Flood, Smurf .
- **Probe**- Grupa ataków polegająca na sprawdzeniu, które usługi są dostępne (przez skanowanie portów TCP/UDP) oraz pod kontrolą, jakiego systemu operacyjnego pracuje badany host.
- **U2R** (*User to Root*) Agresor posiada konto lokalne na atakowanym hoście i próbuje zdobyć uprawnienia konta Root.
- **R2L** (*Remote to Local*) Agresor nie posiada konta lokalnego na atakowanym hoście i dąży do jego zdobycia

Każde połączenie jest opisane za pomocą 41 cech, które możemy podzielić na cztery grupy:

- od 1 do 9 (*Basic Features*) związane z podstawowymi parametrami połączenia IP zawartymi w nagłówkach pakietów.
- od 10 do 22 (*Content Features*) związane z treścią przesyłanych danych i służące głównie do wykrywania ataków typu R2L i U2R

- od 23 do 31 (*Time-based Traffic Features*) związane z transmisją danych i obliczone dla okna czasowego równego 2 sekundy
- od 32 do 41 (*Host-based Traffic Features*) związane z transmisją danych i obliczone dla okna zawierającego 100 sąsiadujących w czasie połączeń do tego samego hosta docelowego. Uwzględniają one ataki, które mogą zachodzić w oknie czasowym większym niż 2 sekundy. Są to na przykład ataki polegające na skanowaniu portów co minutę.

W tabeli 3 w załączniku zostały przedstawione wszystkie cechy wchodzące w skład każdej z grup oraz ich definicje. Struktura podziału danych KDD 99 została przedstawiona w tabeli 2.

Dataset	DoS	Probe	U2r	U2l	Normal
10%KDD	391458	4107	52	1126	97277
Corrected KDD	229853	4166	70	16347	60593
Whole KDD	3883370	41102	52	1126	972780

Tabela 2. Struktura danych KDD99 [2]

- ***10%KDD*** Plik 10% KDD zawiera dane przeznaczone do nauki wybranego systemu wykrywania włamań. Dane należące do poszczególnych połączeń są etykietowane jako normalny ruch lub jako atak. Tylko 10% danych z całego zebranego zbioru (*whole KDD*) zostało przeznaczonych do nauki. Połączenia zebrane w zbiorze KDD10% obejmują 22 typy ataków takich jak: back (Dos), buffer_overflow (u2r), ftp_write (r2l), guess_passwd (r2l), imap (r2l), ipsweep (probe), land (Dos), loadmodule (u2r), multihop (r2l), neptune (Dos), nmap (probe), perl (u2r), phf (r2l), pod (Dos), portsweep (probe), rootkit (u2r), satan (probe), smurf (Dos), spy (r2l), teardrop (Dos), warezmaster (r2l), warezclient (r2l). Poszczególne ataki nie są reprezentowane tą samą liczbą połączeń. Większość danych to ataki wchodzące w skład grupy DoS, co wynika z charakteru tego typu ataków bazujących na dużej liczbie pakietów.

- **Corrected KDD** Zbiór „Corrected" jest używany do testowania już nauczonego sytemu IDS. Zawiera on dodatkowe 14 typów ataków, których nie ma w zbiorze whole KDD oraz 10%KDD. Pozwala to na zbadania jak radzi sobie testowany system z typami ataków, które nie były prezentowane w procesie nauki. W zbiorze znajdują się symulacje następujących ataków nie zawartych w 10%KDD: snmpgetattack, named (r2l), xlock (r2l), xsnoop (r2l), sendmail (r2l), saint (probe), xterm (u2r), mscan (probe), processtable (Dos), ps (u2r), appache2 (Dos), udpstorm (Dos), httptunnel (r2l), worm., mailbomb (Dos), sqlattack., snmpguess.

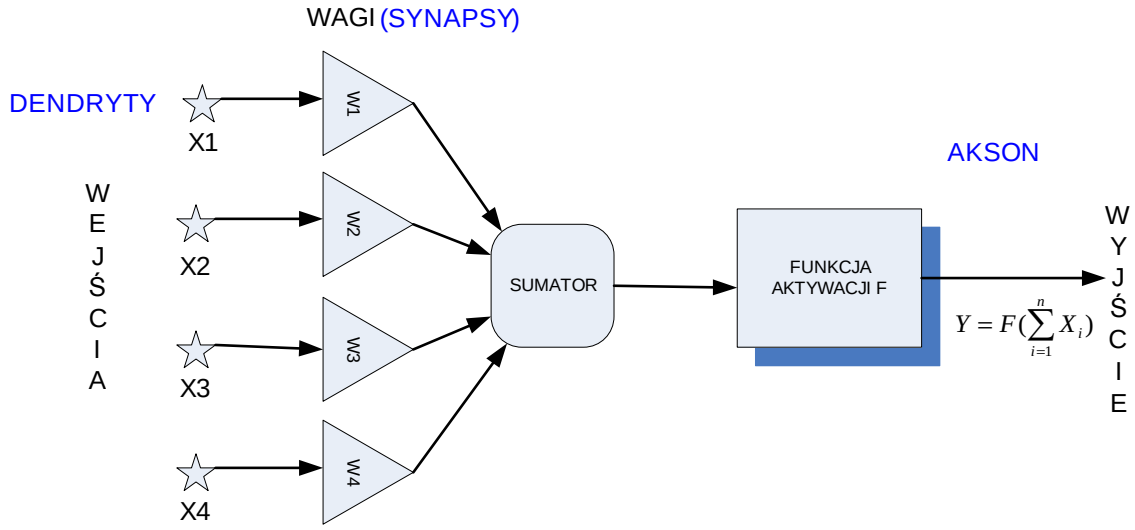
Konkurs KDD został wygrany przez Bernharda Pfahringera z „*Austrian Institute for Artificial Intelligence*", którego klasyfikator bazował na drzewach decyzyjnych, drugie miejsce zajęło narzędzie *Kernel Miner* stworzone przez Itzhaka Levina z LLSoft również bazujące na koncepcji drzew decyzyjnych. Szczegóły dotyczące wyników konkursu można znaleźć w [6].

5. Przykłady zastosowania danych przygotowanych w projekcie KDD

Jako narzędzia służące do wykrywania ataków bazując na danych KDD było wykorzystywanych wiele systemów sztucznej inteligencji takich jak: drzewa decyzyjne, sieci neuronowe, algorytmy genetyczne, systemy rozmyte oraz technologia wektorów wspierających SVM (*Support Vector Machine*).
Poniżej przytoczone krótkie definicje wybranych dwóch systemów bazujących na sztucznej inteligencji: sieci neuronowych oraz drzew decyzyjnych**.**

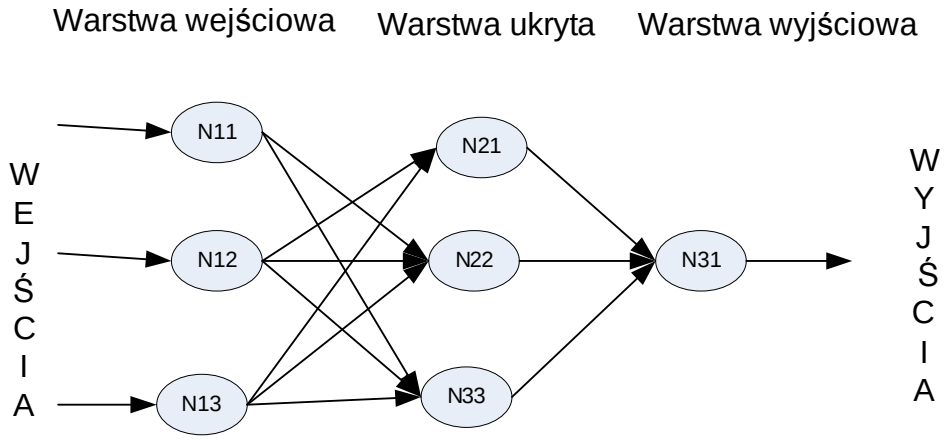
5.1 Sieci neuronowe

Sieci neuronowe są to systemy mające na celu symulowanie pracy neuronów pracujących w obrębie ludzkiego mózgu. Na rysunku 2A przedstawiono schemat działania takiego sztucznego neuronu.



Rysunek 2A. Schemat działania sztucznego neuronu

Składa się on pewnej liczby wejść odwzorowujących dendryty z rzeczywistego neuronu, modułu sumatora (w rzeczywistym mózgu synapsy), funkcji aktywacji oraz jednego wyjścia (reprezentującego akson z ludzkiego mózgu). Sygnały wejściowe x są mnożone przez wartości wag, a następnie wyniki owego mnożenia dodawane są do siebie w bloku sumatora. Otrzymana suma jest wysyłana do bloku aktywacji, gdzie zostaje przetworzona za pomocą określonej funkcji aktywacji. W ten sposób powstaje odpowiedź neuronu „y" na sygnały wejściowe „x".



Rysunek 2B. Schemat trójwarstwowej sieci neuronowej

Siec neuronowa składa się z dużej liczby takich neuronów często położonych w kilku warstwach (rys 2b). Zanim będzie ona wykorzystane do rozwiązania określonego problemu powinna być nauczona jak sobie z nim radzić. W procesie nauki modyfikacji podlegają wagi każdego neuronu aż do momentu osiągnięcia możliwie małej wartości funkcji celu reprezentującej różnice pomiędzy pożądana a otrzymana odpowiedzią sieci na określony wektor wejściowy. Można wyróżnić dwa rodzaje nauki:

- z nauczycielem polegająca na podaniu jakiego rodzaju odpowiedzi oczekujemy od sieci dla określonego wektora wejściowego. Przykładem takiej sieci jest wielowarstwowa siec perceptronowa (MLP).
- Bez nauczyciela gdzie nie podajemy, jaka jest pożądana odpowiedź sieci i pozostawiamy jej właściwe pogrupowania danych. Przykładem takiej sieci jest sieć samoorganizująca się SOM. Nauka takiej sieci polega na tym, że dla określonej grupy wektorów wejściowych wagi jednego z neuronów są najbardziej

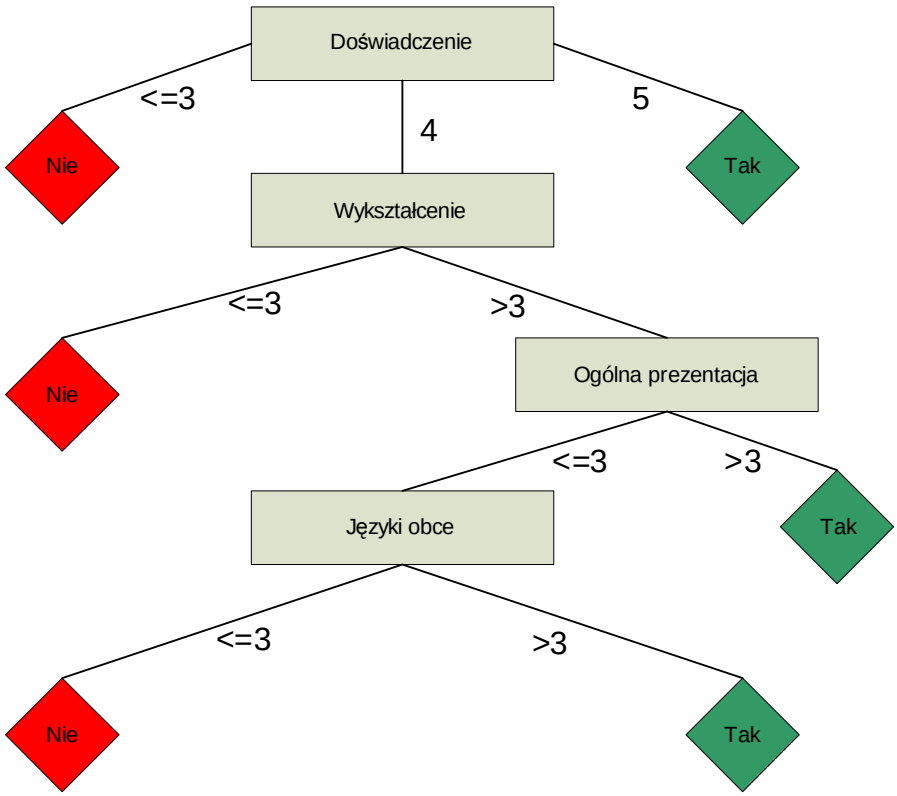
zbliżone do wektora wejściowego i tylko ich wartości oraz wartości wag neuronów w odpowiednio zdefiniowanym sąsiedztwie podlegają modyfikacji w dalszym procesie nauki.

5.2 Drzewa decyzyjne

Drzewo decyzyjne jest grafem o strukturze drzewiastej, którego wierzchołek opisany jest przez pewien atrybut, natomiast poszczególne gałęzie reprezentują możliwe wartości tego atrybutu. Decyzja, do której gałęzi ma być przesłany dany obiekt jest podejmowana na podstawie prostych testów. Węzły drzewa na niższych poziomach są przyporządkowane kolejnym atrybutom, natomiast na najniższym poziomie otrzymujemy liście informujące o przynależności obiektu do określonej klasy czy też podjecie konkretnej decyzji. Przykład takiego drzewa reprezentujący zbiór jest pokazany na rysunku 3B. Celem utworzonego tu prostego drzewa decyzyjnego jest podjęcie decyzja o przyjęciu kandydata na praktyki. Pierwszy krok to stworzenie tablicy decyzyjnej (rysunek 3B) zawierającej wartości poszczególnych atrybutów na podstawie, której będzie budowane drzewo. Rolę atrybutów w prezentowanym przykładzie pełnia: Wykształcenie, Języki obce, Doświadczenie i Ogólne wrażenie mogące przyjmować wartości od 1 do 5. Liście to decyzje o przyjęciu kandydata „tak” lub „nie”.

Inicjały	Wykształcenie	Języki obce	Doświadczenie	Ogólna prezentacja	Przyjęty
MM	2	4	1	4	nie
KL	4	3	4	2	nie
JF	4	5	5	4	tak
GH	1	3	2	3	nie

Rysunek 3A.Przykładowy schemat budowy drzewa decyzyjnego- tabela decyzyjna [12]

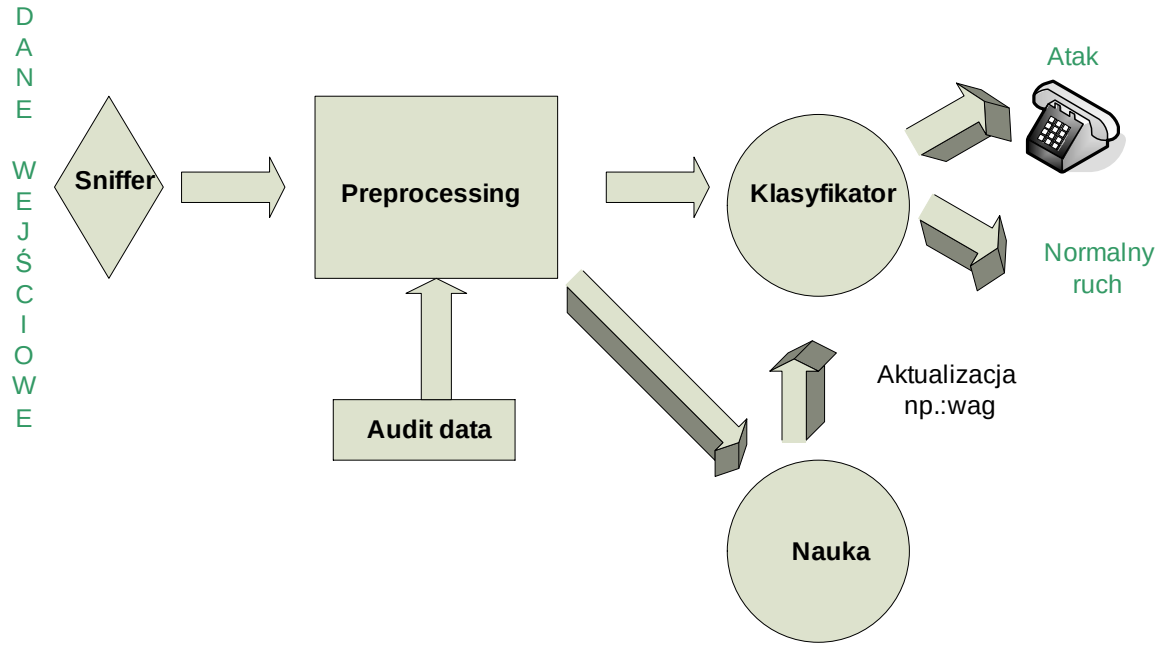


Rysunek 3B. Przykładowy schemat budowy drzewa decyzyjnego- wygenerowane drzewo [12]

Klasycznym algorytmem drzewa decyzyjnego jest ID3. Natomiast jego bardziej rozbudowaną odmianą pozwalająca miedzy innymi na zapobieganie nadmiernemu rozrostowi danych czy obsługę danych wejściowych z brakującymi wartościami dla atrybutów jest algorytm C4.5.

5.3 Wybrane przykłady zastosowania.

Na rysunku 4 przedstawiono bardzo ogólny schemat blokowy systemu bazującego na sztucznej inteligencji.



Rysunek 4. Schemat blokowy systemu IDS

Dane wejściowe są pobrane z sieci IP za pomocą sniffera bądź mogą pochodzić z audytu logów systemu operacyjnego. Są następnie przetwarzane w bloku preprocessingu do postaci przyswajalnej przez moduł klasyfikatora. Role klasyfikatora może pełnić wiele różnych systemów wykorzystujących techniki bazujące na sztucznej inteligencji. Bardzo często w badaniach były wykorzystywane tu sieci neuronowe bądź drzewa decyzyjne.

Mehdi Moradi i Mohammad Zulkernine przedstawili koncepcję zastosowania sieci neuronowej MLP (*multi layer perceptron*)jako klasyfikatora do wykrywania włamań [7]. Wykorzystywana sieć pozwala na określenie, jakiego rodzaju atak nastąpił, dzięki czemu możliwe jest podjęcie właściwej akcji zaradczej. Klasyfikator przedstawiony w pracy rozpoznaje dwa typy ataków: satan i neptune oraz ruch związany z normalna aktywnością użytkownika. Do testów i nauki został wybrany zbiór zawierający około 15 000 połączeń. Przy zastosowaniu dwóch warstw ukrytych uzyskano dokładność klasyfikacji na poziomie 91% natomiast przy jednej warstwie ukrytej około 87%.

W pracy [8] przedstawiono wykorzystanie sieci neuronowej opartej na zasadzie samoorganizacji SOM (*Self Organizing Maps*). Sieć tam przedstawiona jest złożona z dwóch niezależnych warstw SOM, a jako dane wejściowe zostały podane wstępnie przetworzone połączenia reprezentowane tylko przez 6 pierwszych podstawowych cech z zestawu opisanego w tabeli 3.

Podczas gdy ataki z grupy *Probing* i DoS były w większości wykrywane duże problemy stwarzały ataki bazujące na zawartości przenoszonych w sieci danych: R2L i U2R. W pracy [9] zaproponowano hybrydowy system złożony z sieci neuronowej MLP (*multi layer perceptron*) i klasycznego algorytmu drzew decyzyjnych C4.5. Pierwszy składnik jest bardziej skuteczny dla ataków z grup *Probing* i Dos natomiast drugi lepiej radzi sobie z wykrywaniem ataków u2r i r2l. Do testów tego rozwiązania wykorzystano dane zawierające po jednym ataku z każdej grupy oraz dane reprezentujące normalne zachowanie użytkownika. Osiągnięto dokładność wykrywania ataków na poziomie 93,28% przy fałszywych alarmach na poziomie 0,2 %.

W celu poprawienia wydajności systemów IDS w pracy [10] zaproponowano wybór najistotniejszych cech dla każdej grupy ataków. Do tego celu wykorzystano algorytm FNT (*Flexible Neural Tree*). Jego zastosowanie pozwala na zredukowanie danych wejściowych dla poszczególnych klas do 4 (normal), 12 (probe),

12 (Dos), 8 (u2r) i 10 (r2l).W pracy przedstawiono również wyniki symulacji, w której osiągnięto dokładność wykrywania włamań na poziomie 98,39 % do 99,7 % przy ilości fałszywych alarmów na poziomie od 0,1 do 0,8 %.

6. Podsumowanie

Program DARPA dostarczył danych wejściowych umożliwiających testowanie skuteczności działania systemów IDS. Stworzenie ogólnie dostępnego zbioru danych zawierających symulowane ataki oraz normalny ruch pozwala na obiektywne porównywanie nowo tworzonych systemów IDS w tych samym warunkach i za pomocą tych samych informacji wejściowych. Powstały na podstawie programu DARPA projekt KDD dzięki wyodrębnieniu cech charakteryzujących ruch sieciowy i sygnatury ataków ułatwił prace nad zastosowanie systemów sztucznej inteligencji do wykrywania włamań.

Wielu innych badaczy oprócz przedstawionych w rozdziale piątym wykorzystywało zbiór KDD w swoich pracach. Dokładność wykrywania ataków oscylowała w większości przypadków na poziomie ponad 90% przy mniejszej niż 1% ilości fałszywych alarmów. Ponieważ powstał on około osiem lat temu na pewno wymaga uaktualnienia o nowe typy ataków. Dodatkowo pewne cechy mają małe znaczenie dla niektórych typów ataków i mogłyby być wyeliminowane w celu zbudowania wyspecjalizowanego w wykrywaniu mniejszej grupy ataków, ale jednocześnie bardziej wydajnego systemu IDS. Niemniej jednak zbiór KDD jest dobrym wejściowym materiałem do prac nad nowymi bardziej skutecznymi systemami wykrywania włamań.

7. Załącznik

lp	*Nazwa cechy*	*Opis*
1	duration	Czas trwania połączenia w sekundach
2	protocol_type	Typ protokołu: tcp, udp, icmp
3	service	Usługa sieciowa, do której skierowane jest połączenie, np.:, http, telnet, ftp etc.
4	src_bytes	Ilość danych przesłanych ze źródła do miejsca przeznaczenia prezentowana w bajtach
5	dst_bytes	Ilość danych przesłanych z miejsca przeznaczenia do źródła prezentowana w bajtach
6	flag	Status połączenia: normalny lub błąd
7	land	1 jeżeli połączenie pochodzi z tego samego hosta/portu co docelowy; 0 w innym przypadku
8	wrong_fragment	Ilość błędnych fragmentów
9	urgent	Ilość pakietów z ustawioną flagą pilności ”*urgent*”
10	hot	Ilość wskaźników ``hot'' świadczących o np. próbie dostępu do systemu katalogów, utworzeniu pliku wykonywalnego, uruchomieniu programu
11	num_failed_logins	Ilość prób logowania zakończonych porażką
12	logged_in	1 w przypadku próby logowania zakończonej sukcesem; 0 w przeciwnym wypadku
13	num_compromised	Ilość stanów „kompromitujących” w docelowym hoście np.: file/path „not found” error, instrukcja jump to
14	root_shell	1 jeżeli powłoka root została osiągnięta; 0 w przeciwnym przypadku
15	su_attempted	1 jeżeli została wykonana komenda ``su root'' ; 0 w przeciwnym wypadku
16	num_root	Ilość dostępów do konta ``root''
17	num_file_creations	Ilość operacji tworzenia nowych plików

18	num_shells	ilość znaków zachęty powłoki
19	num_access_files	Ilość operacji dostępu do plików odpowiedzialnych za kontrole dostępu np etc/passwd lub .rhosts
20	num_outbound_cmds	Ilość komend poza pasmem w sesji ftp
21	is_hot_login	1 jeżeli login należy do listy ``hot' (np.:root, adm) ; 0 w przeciwnym wypadku
22	is_guest_login	1 w przypadku użycia loginu ``guest" (np.: guest, anonymous); 0 w przeciwnym wypadku

23	count	Ilość połączeń do tego samego hosta w odniesieniu do bieżącego analizowanego połączenia, które wystąpiły w ciągu ostatnich 2 sekund
		Poniższe cechy dotyczące tego samego hosta
24	serror_rate	% błędnych połączeń z ustawioną flagą SYN (oznaczenie S0 w danych).
25	rerror_rate	% błędnych połączeń odrzuconych jako REJ
26	same_srv_rate	% połączeń do tej samej usługi
27	diff_srv_rate	% połączeń do różnych usług
28	srv_count	Ilość połączeń skierowanych do tej samej usługi co bieżące analizowane połączenia które wystąpiły w ciągu ostatnich 2 sekund.
		Poniższe cechy dotyczą tej samej usługi
29	srv_serror_rate	% błędnych połączeń z ustawioną flagą SYN
30	srv_rerror_rate	% błędnych połączeń odrzuconych jako REJ
31	srv_diff_host_rate	% połączeń do różnych hostów
32	dst host count	Liczba połączeń skierowanych do tego samego hosta docelowego
33	dst host srv cont	Liczba połączeń skierowanych do tego samego hosta i dotyczących tej samej usługi
34	dst host same srv rate	% połączeń skierowanych do tego samego analizowanego hosta i dotyczących tej samej usługi
35	dst host diff srv rate	% różnych usług pracujących w ramach danego hosta
36	dst host same src port rate	% połączeń do analizowanego hosta posiadających ten sam numer źródłowy portu
37	dst host srv diff host rate	% połączeń do tej samej usługi pochodzących od różnych hostów
38	dst host serror rate	% błędnych połączeń do tego samego hosta z ustawioną flagą SYN (błąd S0)
39	dst host srv serror rate	%% błędnych połączeń do tego samego hosta i określonej usługi które mają ustawioną flagą SYN (błąd S0)
40	dst host terror rate	% połączeń do danego hosta które mają błąd RST
41	dst host srv terror rate	% połączeń do danego hosta oraz określonej usługi która mają błąd RST

Tabela 3. Cechy wchodzące w skład pojedynczego wektora danych KDD99 [1]

Literatura:

[1]W. Lee, S.J. Stolfo, "A Framework for Constructing Features and Models for Intrusion Detection Systems", ACM Transactions on Information and System Security (TISSEC), 3(4):227--261, 2000.

[2] H. G. Kayacik, A. N. Zincir-Heywood, M.I. Heywood, “Selecting Features for Intrusion Detection: A Feature Relevance Analysis on KDD 99 Intrusion Detection Datasets.”, In Proceedings of the Third Annual Conference on Privacy, Security and Trust, St. Andrews, Canada, October 2005,
[3] W. Lee S.J. Stolfo, “Data Mining Approaches for Intrusion Detection”, Preceedings of the Seventh USENIX security Symposium(SECURITY '98), San Antonio, TX,1998
[4]R. Lippmann, J.W. Haines, D.J. Fried, J.Korba, K. Das, “The 1999 Darpa Off-Line Intrusion Detection Evaluation”, Computer Networks: The International Journal of Computer and Telecommunications Networking 34 (2000) 579--595.,2000
[5]V. Paxson,” Bro: A system for Detecting Network Intruder in Real Time” In Proceedings of the 7th USENiX Security Symposium, San Antonio 1998
[6]Charles Elkan, “Results of the KDD’99 Classifier-learning contest”, In `http://www-cse.ucsd.edu/#elkan/clresults.html', September 1999
[7]M. Moradi, M. Zulkernine, “A neural network based system for intrusion detection and classification of attacks”, Proc. of the 2004 IEEE International Conference on Advances in Intelligent Systems - Theory and Applications, pp. 148:1-6, Luxembourg, November 2004.
[8] P. Lichodzijewski A. Nur Zincir-Heywood M. I. Heywood, “Dynamic Intrusion Detection using Self Organizing /maps”,In Processing Annual Canadian Information Technology Security Symposium, May 2002
[9] Z-S. Pan S-C. Chen, G-B. Hu, D-Q Zhang, “Hybrid neural network and c 4.5 for misuse detection”, 2003
[10]Y. Chen; A. Abraham; B. Yang, “Hybrid Flexible Neural-Tree Based IDS”, International Journal of Intelligent Systems, vol. 22, no. 4, pp. 337–352, 2007
[11]http://www.statsoft.pl/textbook/stathome.html
[12]http://pl.wikipedia.org/wiki/Drzewo_decyzyjne