Koncepcja hybrydowego systemu detekcji robaków sieciowych wykorzystującego metody eksploracji danych

Sławomir Mika

Streszczenie: W artykule została zaprezentowana koncepcja hybrydowego systemu wykrywania robaków sieciowych wykorzystującego metody eksploracji danych. Proponowane podejście bazuje na detekcji anomalii w monitorowanym ruchu sieciowym, czyli identyfikowaniu wszelkich odchyleń różnych od wyuczonego, normalnego profilu sieci, które z reguły wskazują na próbę ataku. Prezentowany model systemu analizuje zarówno dane użytkowe niesione przez pakiet, jak również bada nagłówek pakietu.

Słowa kluczowe: eksploracja danych, robak sieciowy, detekcja anomalii

1. WPROWADZENIE

Ataki dotąd nieznanych, nowych robaków sieciowych tzw. *zero-day* stanowią obecnie poważny problem użytkowników sieci komputerowych. Co więcej, duża liczba wzajemnie połączonych i podatnych na to zagrożenie systemów umożliwia ich łatwe rozprzestrzenianie się w sieci. Powszechnie wykorzystane systemy obrony przed zagrożeniami sieciowymi oraz techniki wykrywania ataków opierają swoje działanie na wcześniej ustalonych i znanych sygnaturach tychże ataków. Z tego powodu nie potrafią one skutecznie zapobiec lub wykryć nowy rodzaj ataku, czyli nieznanego wcześniej robaka sieciowego.

Znanych jest wiele przykładów robaków sieciowych, które gwałtownie rozprzestrzeniają się w sieci np. *Code Red*, *Slammer* lub *Conficker*. W konsekwencji bardzo szybko infekują i niszczą kolejne systemy powodując często znaczne straty finansowe.

Istnieją dwa podejścia stosowane w sieciach komputerowych, na których bazują systemy wykrywania i zapobiegania włamaniom [3]. Pierwsze opiera się na wykorzystaniu sygnatur znanych ataków. Bezpieczeństwo w tym przypadku jest głównie uwarunkowane aktualnością wykorzystanej bazy znanych ataków oraz szybkością identyfikowania nowych zagrożeń przez ekspertów, którzy rozpoznają nowe ataki. Główną zaletą tych systemów jest wysoka skuteczność wykrywania znanych zagrożeń. Z drugiej strony nie potrafią one wykrywać nieznanych dotąd ataków. Drugie podejście opiera się na wykorzystaniu technik, które identyfikują ataki bazując na wykrywaniu odchyleń względem ustalonego profilu sieci, a reprezentującego jej typową i oczekiwaną aktywność [3].

Większość systemów bazujących na detekcji odchyleń od ustalonego profilu ruchu sieciowego wykorzystuje techniki eksploracji danych i metody odkrywania wiedzy w celu zwiększenia swojej wydajności i efektywności [14]. Techniki eksploracji danych służą jako skuteczne narzędzie do wydobycia ukrytych zależności w zbiorach danych i dlatego stanowią często rdzeń badań. Innymi słowy dzięki technikom data mining możliwe jest utworzenie szczegółowego opisu sieci. Co więcej, pozwalają one na ujawnienie ukrytych zależności i współzależności, które bez użycia tychże technik byłyby trudne do wykrycia.

W artykule został zaproponowany model hybrydowego sytemu wykrywania robaków sieciowych wykorzystujący metody eksploracji danych tzw. *data mining*. Takie podejście motywowane jest przeświadczeniem, że kombinacja różnych technik eksploracji danych zwiększa finalną skuteczność systemu. Działanie modelu opiera się na początkowym ustaleniu profilu sieci, a następnie w fazie detekcji wykrywaniu wszelkich odchyleń od typowego ruchu sieciowego. Należy zaznaczyć, że w opisywanym przypadku system będzie aplikacją komputerową.

Powszechnie proponowane są dwa sposoby wykorzystania detekcji odchyleń od normalnego profilu ruchu sieciowego. Pierwszy bierze po uwagę nagłówki pakietów oraz wszelkie dodatkowe informacje charakteryzujące ruch sieciowy takie jak znaczniki czasowe, ilość przesłanych danych, szybkość transferu danych, itd. Jednak w ogóle nie bada *payload'u*. Zupełnie przeciwną grupę systemów reprezentuje analiza wyłącznie danych użytkowych niesionych przez pakiety. Obecnie systemy wykrywania intruzów w sieciach komputerowych bazują w zdecydowanej większości tylko na jednym, wybranym podejściu. Koncepcja systemu przedstawiona w tym artykule bazuje na obu wymienionych technikach. W rezultacie oczekuje się zwiększenia dokładności działania systemu. Wynika to z tego, że ataki robaków sieciowych wiążą się z dostarczaniem złych danych, dlatego dokładna analiza danych niesionych przez pakiety jest bardzo ważna. Nie należy jednak zapominać, że robaki sieciowe dążą do rozprzestrzeniania się w sieci, dlatego analiza nagłówków jest równie istotna co analiza danych pakietów. Związane jest to z generowaniem dodatkowego ruchu sieciowego przez propagujące robaki internetowe,

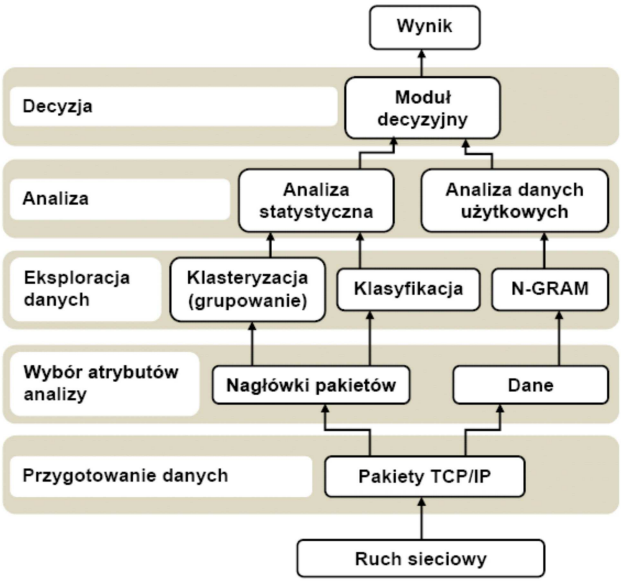
który często nie pokrywa się z ustalonym wcześniej profilem [2].

Idea połączenia kilku klasyfikatorów w jednym modelu, w celu zwiększenia jego efektywności, nie jest podejściem całkowicie nowym. Przykładowo w pracy [5] zaprezentowany jest system opierający się na trzech metodach: naiwnym klasyfikatorze Bayes'a, sieci neuronowej oraz algorytmie k-najbliższych sąsiadów. Mimo to prezentowany w artykule system wyróżnia się ze względu na integracje w obrębie systemu technik analizy nagłówka oraz *payload'u* pakietów. Dodatkowo w celu zwiększenia dokładności wykrywania intruzów łączy on w obszarze analizy nagłówków pakietów dwie grupy technik eksploracji danych tj. grupowanie oraz klasyfikacje.

2. MODEL SYSTEMU

Proponowany model systemu w celu określenia profilu ruchu sieciowego używa zarówno nagłówka jak i danych użytkowych niesionych przez pakiety. Mechanizm wykrywania ataków robaków sieciowych opiera się na wykorzystaniu wcześniej ustalonego profilu ruchu sieciowego. W konsekwencji algorytmy wykrywania nieprawidłowości i odchyleń od typowego profilu sieci pracują w dwóch fazach: fazie uczenia i fazie detekcji.

W fazie uczenia typowy i normalny ruch sieciowy jest obserwowany i odpowiednie techniki uczące, dedykowane właściwym metodom analizy, pozwalają zbudować odpowiedni profil. Z kolei w fazie detekcji, ruch sieciowy jest porównywany z ustalonym wcześniej profilem i każde odchylenie od normalnego zachowania się sieci jest traktowane jako sytuacja nieprawidłowa.



Rys. 1. Model systemu wykrywania ataków robaków sieciowych

Na rysunku 1 przedstawiono model systemu wykrywania ataków robaków internetowych. Ruch sieciowy jest stale monitorowany, a kolejne pakiety IP są filtrowane i wstępnie obrabiane w fazie przygotowania danych. Na tym etapie w ramach każdego pakietu wyodrębniany jest nagłówek i dane użyteczne niesione w części danych pakietu. Następnie podczas wyboru atrybutów, poddawane są one odpowiednim procesom pozwalającym ostatecznie uzyskać konkretne dane, które stanowią dane wejściowe dla odpowiednich technik eksploracji danych. Dane użytkowe pakietu są analizowane z użyciem analizy *n-gram*, która jest typową metodą statystyczną. Z kolei odpowiednie atrybuty nagłówka pakietu przetwarzane są z użyciem technik klasyfikacji oraz metod klasteryzacji (grupowania). Efektem uczenia wybranych technik eksploracji danych, wykorzystywanych zarówno w przypadku nagłówka, jak i danych użytkowych pakietu, jest powstanie odpowiednich modeli określających normalny profil ruchu sieciowego. W fazie detekcji otrzymane uprzednio modele wykorzystywane są w procesach porównania, a następnie ich wynik przekazywany jest do bloku decyzyjnego, który opiera się na modelu probabilistycznym – naiwnym klasyfikatorze Bayes'a. Należy zaznaczyć, że na etapie technik eksploracji danych uczenie klasyfikatorów będzie przebiegało bez nadzoru. Natomiast moduł decyzyjny będzie uczony pod nadzorem.

Ważną cechę proponowanego systemu stanowi możliwość tworzenia profilu sieci w trybie inkrementacyjnym. Umożliwia to łatwe przystosowanie się systemu do zmian pojawiających się w monitorowanej sieci. Profil sieci może być uaktualniony nawet w trakcie fazy detekcji, jeśli zajdzie taka potrzeba. Ostatecznie prezentowany system powinien charakteryzować się zwiększoną precyzja wykrywania robaków sieciowych, dzięki połączeniu i kombinacji metod analizy nagłówka i danych użytkowych pakietów. Ostatnią ważną cechą modelu jest częściowa odporność na specyficzny typ ataku, gdy atakujący próbuje ukryć swoją aktywność poprzez powielanie normalnego zachowania sieci.

W kolejnych rozdziałach zostaną dokładnie opisane algorytmy i techniki wykorzystane w proponowanym systemie wykrywania robaków sieciowych.

3. EKSPLORACJA DANYCH

W trakcie analizy odpowiednich danych zebranych w procesie monitorowania ruchu sieciowego wykorzystywane są różne techniki eksploracji danych. W kolejnych sekcjach rozdziału scharakteryzowane są krótko metody użyte w proponowanym modelu. Sprowadza się to głównie do przedstawienia ogólnego procesu postępowania w przypadku wybranej grupy technik. Wszystkie metody są trenowane podczas procesu uczenia nienadzorowanego, co sprawia że proces uczenia jest całkowicie niezależny od użytkownika. Oznacza to również, że uczenie odbywa się w sposób całkowicie automatyczny. To duży plus proponowanej aplikacji.

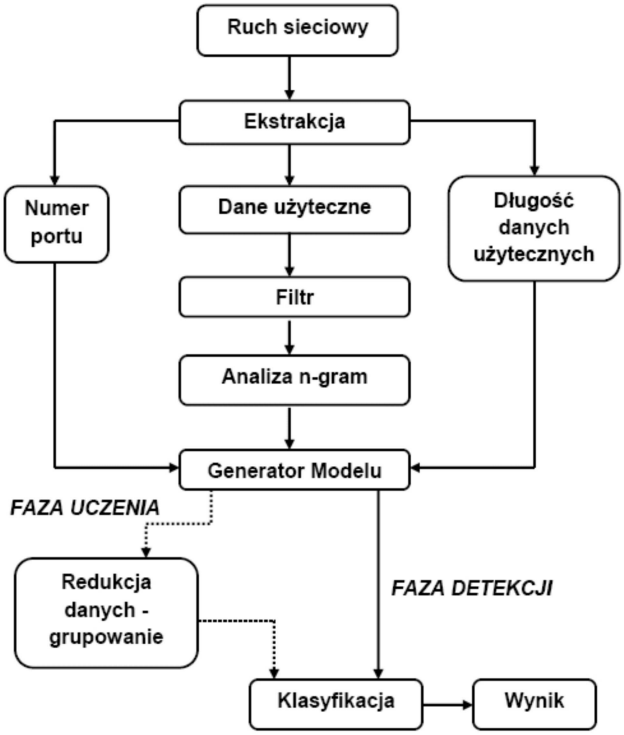
3.1. ANALIZA DANYCH PAKIETU

Schemat bloku analizującego dane użyteczne przenoszone przez pakiety, został przedstawiony na rysunku 2.

Proponowany model został oparty na systemie opisanym w [13]. Wymaga użycia algorytmu analizy *n-gram* oraz wykorzystania klasyfikacji bazującej na grupowaniu pakietów o różnej długości danych użytkowych.

Zastosowanie analizy *n-gram* w omawianym systemie polega na zliczaniu ilości wystąpień konkretnych

wartość bajtów (znaków ASCII) w danych użytecznych przenoszonych przez pakiet. Proponowana metoda zakłada model zbudowany z *uni-gram'ów*, czyli dla $n = 1$. Rezultatem tej analizy jest 256 elementowy wektor określający rozkład częstości bajtów dla każdego pakietu.



Rys. 2. Model bloku analizującego *payload*

Podczas fazy ucznia, dla danego zbioru danych uczących budowane są modele M_{ijk}. Dla każdego pakietu przychodzącego i wychodzącego, dla danego adresu i, dla każdego portu j oraz długości danych użytecznych (*payload'u*) k, model przechowuje zwiększane inkrementacyjnie: średnią częstość wartości bajtów, odchylenie standardowe oraz wariancje. Dla danego modelu M_{ijk} tworzą one 256 elementowe (liczba znaków kodu ASCII) wektory. Przykładową znormalizowaną średnia częstość wartości bajtów dla portu 80 oraz dla długości danych użytecznych równej 1460 przedstawia rysunek 3. Z kolei podczas fazy detekcji te same wartości są obliczane dla każdego pakietu i następnie zostają porównane z wartościami modelu. Wykrycie problemu sygnalizowane jest w sytuacji, jeśli znacząco różnią się od danego modelu. Jako metoda porównania danej próbki danych z modelem jest wykorzystywana uproszczona postać odległości Mahalanobis'a, która jest często używana w zadaniach wymagających komparacji dwóch rozkładów statystycznych. Jej wartość jest obliczana na podstawie wzoru (1):

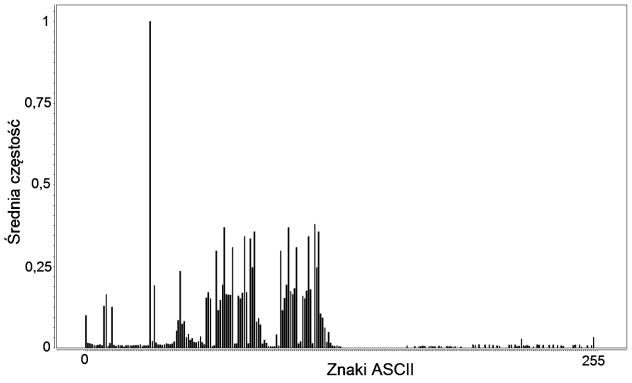
$$d(B,A) = \sum_{i=0}^{255}\left(\frac{|B_i - A_i|}{(\sigma_i + \alpha)}\right), \qquad (1)$$

gdzie B oraz A są wektorami częstości wartości bajtów, wektor B jest wektorem częstości wartości bajtów nowego *payload'u*, a A jest wektorem średnich częstości wartości bajtów obliczonych w procesie uczenia. σ_i jest odchyleniem standardowym dla i-tego znaku ASCII, podczas gdy α jest współczynnikiem wygładzenia dodanym, aby zapobiec możliwości uzyskania nieskończonej wartości odległości dla odchylenia standardowego σ_i równego zero.

Niestety liczba modeli powstałych w procesie uczenia może być bardzo duża ze względu na zróżnicowaną długość pakietów oraz liczbę możliwych portów danego systemu. W związku z tym duża liczba modeli jest redukowana w procesie klasteryzacji (grupowania). Prowadzi to do uporządkowania modeli w klastry, czyli grupy modeli. Elementy danej grupy są podobne bardziej do modeli z danej grupy niż do pozostałych modeli. W procesie porównania dwóch sąsiednich modeli wykorzystana jest odległość Manhattan. Jej wartość jest obliczana na podstawie wzoru (2):

$$d(B,A) = \sum_{i=0}^{255}|B_i - A_i|, \qquad (2)$$

Gdzie B oraz A są wektorami średnich częstości wartości bajtów różnych modeli. Jeśli odległość jest mniejsza niż pewna wartość progowa dwa modele są łączone. Dodatkowo obliczane są nowe wartości średniej rozkładu bajtów, odchylenia standardowego oraz wariancji. Proces grupowania trwa tak długo, aż istnieje jeszcze model możliwy do zgrupowania.



Rys. 3. Znormalizowana średnia częstość wartości bajtów dla portu 80 i długości *payload'u* 1460

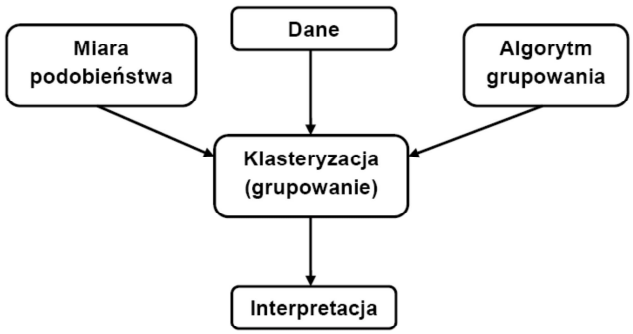
Opisana metoda analizy *payload'u* może zostać dodatkowo zmodyfikowana o użycie filtru Bloom'a [12], który jest prostą i oszczędna pamięciowo strukturą reprezentującą dany zbiór elementów. Innym rozwinięciem prezentowanego modelu może być technika opisana w [6]. Wskazuje ona, że zwiększenie efektywności wykrywania robaków sieciowych można uzyskać porównując dane użytkowe pakietów przychodzących z danymi użytkowymi niesionymi przez pakiety wychodzące. Dlatego proponowany system jest dodatkowo wzbogacony o porównywanie „podejrzanych" modeli opisujących ruch przychodzący i wychodzący dla danego portu. Wiąże się to z faktem, że często robaki internetowe rozprzestrzeniają te same dane, dzięki którym udało im się zainfekować obecny system.

3.2. ANALIZA NAGŁÓWKA PAKIETU

Analiza nagłówka pakietu również polega na ustaleniu odpowiednich profili sieci w fazie uczenia oraz na wykrywaniu wszelkich niezgodności względem tych profili w fazie detekcji. Podstawowymi atrybutami nagłówków, które powinny być uwzględnione w tym procesie są: źródłowy i docelowy adres IP, źródłowy i docelowy numer portu, typ pakietu, długość danych użytecznych przenoszonych przez pakiet oraz czas życia

pakietu (TTL). Wybrane do opisywanego systemu metody reprezentują dwie grupy szeroko stosowanych technik eksploracji danych: klasteryzacji (grupowania) oraz klasyfikacji.

W zastosowaniu wykrywania intruzów metody klasteryzacji wymagają zbioru normalnych danych, które są wykorzystywane w procesie uczenia modelu. W tym przypadku w fazie detekcji, jako anomalie traktowane są wszelkie próbki odstające od wzorca i nie pasujące do utworzonego modelu. Elementy odstające zwane są z angielskiego *outlier'ami*. W celu ich wykrycia stosuje się różne techniki detekcji elementów odstających. Przykładowo można zastosować podejście najbliższego sąsiada, odległości Mahalanobis'a lub oceny gęstości możliwości wystąpienia elementów odstających [7].



Rys. 4. Ogólny model metod grupowania

Ogólny schemat obrazujący działanie algorytmów klasteryzacji jest przedstawiony na rysunku 4. Klasteryzacja danych przeprowadzana jest z wykorzystaniem wybranej miary podobieństwa oraz algorytmu grupowania. Podstawowymi zaletami grupowania są łatwość użycia w systemach pracujących w trybie *on-line* oraz uczenie bez nadzoru. Natomiast główną wadą jest stosunkowo duża złożoność obliczeniowa, szczególnie w procesie uczenia oraz problemy z grupowaniem danych o dużej ilość wymiarów. Proponowane metody grupowania to algorytmy: k-średnich, maksymalnej wartości oczekiwanej (EM), Cobweb lub DBSCAN.

Przykładem dobrze znanej metody eksploracji danych, a często wykorzystywanej w przypadku klasteryzacji jest wspomniany wcześniej algorytm k-średnich. Zastosowany w omawianym systemie pozwala na wykrycie nietypowego nagłówka pakietu, sygnalizując jednocześnie nietypowy ruch sieciowy. Bazuje na wykorzystaniu miary odległości euklidesowej. W procesie uczenia algorytm iteruje przez wszystkie dane uczące, uaktualniając architekturę wewnętrznych grup (klastrów). Każdy wektor x należący do zbioru uczącego może być reprezentowany jako wektor n elementowy (3):

$$x = (x_1, x_2, ..., x_n). \tag{3}$$

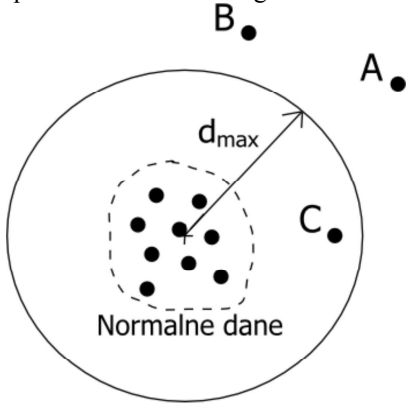
Odległość euklidesowa dwóch wektorów x oraz y wyraża się poniższym równaniem (4):

$$d(x, y) = |x - y| = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}. \tag{4}$$

Natomiast średnią μ zbioru wektorów c_j definiuje wzór (5):

$$\mu = \frac{1}{|c_j|} \sum_{x \in c_j} x. \tag{5}$$

Równanie (5) pozwala wyznaczenie środków każdego klastra. Po wyznaczeniu środków wszystkich k grup algorytm jest gotowy do klasyfikacji. W przypadku omawianego systemu podczas tego procesu wykorzystywane są techniki wykrywania *outlier'ów*. Ich wykrycie będzie oznaczało, że dany pakiet nie jest typowym dla konkretnego, wcześniej wytrenowanego profilu sieci. W tym celu należy obliczyć odległość nowych danych od środka najbliższego klastra korzystając z równania (4). Jeśli obliczona odległość jest większa od wcześniej ustalonej wartości progowej d_{max} oznacza to, że dany pakiet jest traktowany jako *outlier* [10]. Metoda wykrywania *outlier'ów* została zobrazowana na rysunku 5. Obiekty A i B są *outlier'ami*, natomiast obiekt C znajduje się w obszarze ograniczonym przez odległość d_{max} i jest identyfikowany jako, należący do danych reprezentujących nagłówki pakietów typowe dla ustalonego profilu ruchu sieciowego sieci.

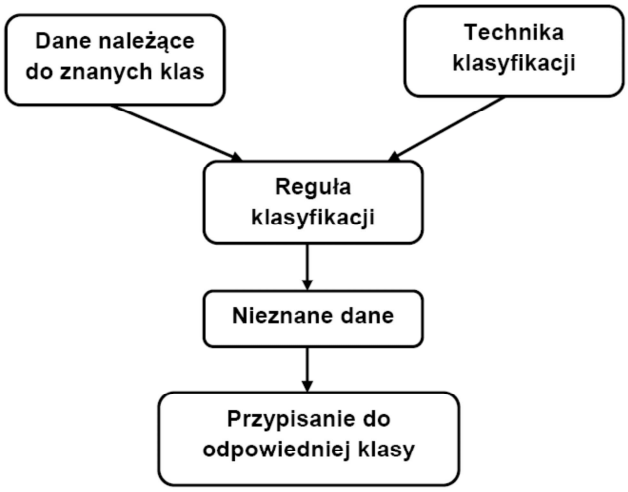


Rys. 5. *Wykrywanie outlier'ów*

Druga grupę technik eksploracji danych stanowią metody klasyfikacji. W przeciwieństwie do technik klasteryzacji uczenie zachodzi w procesie pod nadzorem przy czym odpowiednie modele są uczone na treningowej próbie ruchu sieciowego, przy założeniu, że analizowane dane uczące są wolne od wszelkich odchyleń i anomalii. Wobec uczenia modeli jedynie na normalnym i typowym ruchu sieciowym należy zastosować odpowiednie techniki uczenia zwane *PU learning* [8]. Umożliwiają one uczenie w sytuacji, gdy do procesu ucznia możliwe jest wykorzystanie tylko pozytywnych danych uczących. Doskonale nadają się one w prezentowanym modelu ponieważ są całkowicie automatyczne i nie wymagają posiadania w procesie uczenia danych oznakowanych jako infekcje lub ataki. W literaturze istnieje wiele podejść do problemu uczenia PU learning. Jedno z nich jest opisane w [8], gdzie zaproponowane jest dwu stopniowe podejście do procesu uczenia klasyfikatora.

Ogólny model obrazujący proces klasyfikacji jest przedstawiony na rysunku 6. Identyfikowanie danych przeprowadzane jest dzięki wybraniu odpowiedniej techniki klasyfikacji oraz wstępnej znajomości przynależności danych do określonych klas. W efekcie zastosowania danej reguły klasyfikacji, możliwe jest przypisanie nieznanych (nowych) danych do odpowiedniej klasy. Podstawową zaletą grupowania jest wysoka skuteczność i dokładność procesu uczenia. Z kolei

najważniejszą wadą okazuje się być możliwa duża liczba fałszywych alarmów. Proponowane techniki klasyfikacji to algorytmy k-najbliższych sąsiadów oraz C4.5 (drzewa decyzyjne).



Rys. 6. Ogólny model metod klasyfikacji

4. NAIWNY KLASYFIKATOR BAYES'A

Moduł decyzyjny stanowi niejako ostatnią warstwę proponowanego systemu (rys. 1). Dane wejściowe tego bloku stanowią rezultaty analizy nagłówka i danych użytecznych pakietów. Natomiast informacja wyjściowa z modułu stanowi ostateczny rezultat analizy ruchu sieciowego.

Zastosowany naiwny klasyfikator Bayes'a jest prostym klasyfikatorem probabilistycznym bazującym na modelach probabilistycznych. Uzyskiwane są one wskutek użycia reguły Bayes'a. Duża zaletą naiwnego klasyfikatora Bayes'a jest wysoka efektywność uczenia w procesie nadzorowanym. Pomimo naiwnej budowy oraz na pozór zbyt uproszczonych założeń, naiwny klasyfikator Bayes'a bardzo często pracuje bardzo wydajnie i jest wykorzystywany w wielu praktycznych sytuacjach [4]. W przeciwieństwie do niższych warstw systemu nie jest to moduł o dużym skomplikowaniu. Jego głównym celem jest jedynie interpretacja wyników analizy przeprowadzonej z użyciem omówionych wcześniej technik *data mining*.

Reguła Bayes'a umożliwia obliczenie prawdopodobieństwa hipotezy bazując na jej prawdopodobieństwie a *priori*. Wyraża się ona wzorem (6):

$$P(Y \mid X) = \frac{P(X \mid Y)P(Y)}{P(X)} \tag{6}$$

W równaniu (6) Y oznacza pewną hipotezę, podczas gdy X oznacza wszelkie dane oraz wstępną wiedzę, która może wpłynąć na ocenę prawdopodobieństwa tej hipotezy. $P(Y|X)$ jest prawdopodobieństwem a *posteriori* hipotezy Y przy znajomości danych X, $P(Y)$ jest prawdopodobieństwem a *priori* hipotezy Y, $P(X)$ jest prawdopodobieństwem danych X, a $P(X|Y)$ określa prawdopodobieństwo warunkowe [11].

Naiwny klasyfikator Bayes'a pozwala na obliczenie prawdopodobieństwa warunkowego przy założeniu, że atrybuty danej klasy oznaczonej jako y są warunkowo niezależne od siebie. Założenie warunkowej niezależności przedstawia równanie (7):

$$P(X \mid Y = y) = \prod_{i=1}^{d} P(X_i \mid Y = y), \tag{7}$$

gdzie każdy zbiór atrybutów $X = \{X_1, X_2, \ldots, X_d\}$ składa się z d atrybutów [11].

Podstawową zadaniem podczas procesu budowania naiwnego klasyfikatora Bayes'a jest jego odpowiednie uczenie. Należy założyć, że Y oraz atrybuty $X_1 \ldots X_d$ są reprezentowane jako wartości dyskretne. W tym przypadku prawdopodobieństwo, że Y przyjmie swoją k-tą możliwą wartość, zgodnie z regułą Bayes'a wyraża poniższy wzór (8):

$$P(Y = y_k \mid X_1 \ldots X_d) = \frac{P(Y = y_k)P(X_1 \ldots X_d \mid Y = y_k)}{\sum_j P(Y = y_j)P(X_1 \ldots X_d \mid Y = y_j)}, \tag{8}$$

gdzie suma jest dla wszystkich możliwych wartości y_j danej Y [1,9]. Przyjmując, że atrybuty X_i są warunkowo niezależne, a zatem wykorzystując równie (7), wyrażenie (8) przyjmuje następującą postać (9):

$$P(Y = y_k \mid X_1 \ldots X_d) = \frac{P(Y = y_k)\prod_i P(X_i \mid Y = y_k)}{\sum_j P(Y = y_j)\prod_i P(X_i \mid Y = y_j)}. \tag{9}$$

Równanie (9) jest fundamentalnym równaniem opisującym naiwny klasyfikator Bayes'a. Często istotne jest znalezienie jedynie najbardziej prawdopodobnej wartości Y. Wyraża to reguła (10):

$$Y \leftarrow \arg\max_{y_k} \frac{P(Y = y_k)\prod_i P(X_i \mid Y = y_k)}{\sum_j P(Y = y_j)\prod_i P(X_i \mid Y = y_j)}. \tag{10}$$

Uwzględniając, że mianownik nie jest zależny od y_k równie (10) sprowadza się do postaci (11) tzw. naiwnej reguły klasyfikacji Bayes'a [9]:

$$Y \leftarrow \arg\max_{y_k} P(Y = y_k)\prod_i P(X_i \mid Y = y_k). \tag{11}$$

5. PODSUMOWANIE

Obecnie proponowany hybrydowy system wykrywania ataków robaków sieciowych, a wykorzystujący techniki eksploracji danych jest w fazie budowy. Jednak argumenty przedstawione w artykule pozwalają przypuszczać, że proponowany system wykaże się wysoką skutecznością i precyzją w wykrywaniu robaków internetowych. Opisany model charakteryzuje się stosunkowo małą złożonością w fazie detekcji, a algorytmy w większości charakteryzują się liniową złożonością, dlatego powinien sprawdzić się w rzeczywistych sieciach komputerowych jako system alarmujący o atakach. Szczególną zaletą oferowaną przez system jest wykrywanie nieznanych dotąd robaków sieciowych. Kolejną ważną cechą systemu okazuje się fakt, że faza uczenia oraz faza detekcji trzonu aplikacji bazującego na technikach eksploracji danych, przebiega całkowicie automatycznie. Co więcej system może być trenowany w każdej sieci, przy założeniu, że przez czas uczenia ruch sieciowy będzie wolny od wszelkich nieprawidłowości.

LITERATURA

[1] Caruana R., Niculescu-Mizil A., „An Empirical Comparasion of Supervised Learning Algorithms", *Cornell University*, 2006.

[2] Cheema F. M., Akram A., Iqbal Z., "Comparative Evaluation of Header vs. Payload based Network Anomaly Detectors", *Proceedings of the World congress on Engineering, VOL.1 – WCE 2009*, 2009.

[3] Dokas P., Ertoz L., Kumar V., Lazarevic A., Srivastava J., Tan P. N., "Data mining for network intrusion detection", *The NSF Workshop on Next Generation Data Mining*, 2002.

[4] Farid D., Rahman M. Z., "Anomaly Network Intrusion Detection Based on Improved Self Adaptive Bayesian Algorithm", *Journal of computers, VOL.5*, 2010.

[5] Kholfi S., Habib M., Aljahdali S., "Best hybrid classifiers for intrusion detection", *Journal of Computational Methods in Science and Engineering*, 2006, ss. 299–307.

[6] Kim H. A., Karp B., "Autograph: Toward Automated Distributed Worm Signature Detection", *USENIX Security Symposium*, 2004.

[7] Lazarevic A., Ertoz L., Kumar V., Ozgur A., Srivastava J., "A comparative study of anomaly detection schemes in network intrusion detection", *Proceedings of the Third SIAM International Conference on Data Mining*, 2003.

[8] Li X., Yu P. S., Liu B., Ng S., "Positive Unlabeled Learning for Data Stream Classification", *SDM 2009*, 2009, ss. 257-268.

[9] Mitchell T. M., *Machine learning*, McGraw Hill, 2005.

[10] Münz G., Li S., Carle G., „Traffic Anomaly Detection Using K-Means Clustering", *In GI/ITG Workshop MMBnet*, 2007.

[11] Tan P., Steinbach M., Kumar V., *Introduction to data mining*, Boston: Pearson Addison Wesley, 2006.

[12] Wang K., Parekh J. J., Stolfo S., "Anagram: A Content Anomaly Detector Resistant To Mimicry Attack", *Proceedings of the 9th International Symposium on Recent Advances in Intrusion Detection, RAID*, 2006.

[13] Wang K., Stolfo S., „Anomalous payload-based network intrusion detection", *Recent Advances in Intrusion Detection, RAID*, 2004, ss. 203–222.

[14] Zhang J., Zulkernine M., "Network Intrusion Detection using Random Forests", *IEEE Transactions on Systems, Man, and Cybernetics*, 2008.

mgr inż. Sławomir Mika
Politechnika Łódzka
Katedra Mikroelektroniki i Technik Informatycznych

ul. Wólczańska 221/223
90-924 Łódź

tel.: 663 488 475
e-mail: mika@dmcs.pl