ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 2013
Seria: ORGANIZACJA I ZARZĄDZANIE z. 64 Nr kol. 1894

Jacek GRUBER
Ireneusz J. JÓŹWIAK
Łukasz MOSIO
Politechnika Wrocławska
Wydział Informatyki i Zarządzania

ZASTOSOWANIE EKSPLORACJI DANYCH DO WYKRYWANIA NADUŻYĆ W SYSTEMACH BIZNESOWYCH

Streszczenie. Artykuł zawiera opis przykładu użycia metod eksploracji danych do wykrywania nadużyć w systemach biznesowych IT. Przedstawiono wybrane metody eksploracyjne – k-najbliższych sąsiadów, drzewa wzmacniane, sieci neuronowe, drzewa CHAID, drzewa C&RT, MARSplines. Metody te zastosowano do zbudowania projektu eksploracyjnego w programie STATISTICA 10 PL. Do badań użyto danych z konkursu KDD Cup 99. Zawierają one symulowane ataki w komputerowej sieci wojskowej.

FRAUD DETECTION BUSINESS SYSTEMS USING DATA MINING METHOD

Summary. This paper describes how to use the method of data mining methods to detect fraud. Several useful for fraud detection data mining methods described in this article, namely: k-nearest neighbor, wood-reinforced, neural networks, trees, CHAID, C & RT tree, MARSplines. These methods have been used to build the project exploration in STATISTICA 10 PL. Usefulness of these methods to detect attacks on computer network. The study used data from the KDD Cup 99 competition. These data include simulated attacks on military networks.

1. Wstęp

Eksploracja danych jest coraz bardziej popularną metodą wykorzystują moc obliczeniową komputerów do znajdowania zależności w danych. Znajdywanie takich zależności jest zadaniem dość trudnym, ponieważ ręczna analiza dużych ilości danych jest bardzo czasochłonna i niezwykle trudna do precyzyjnego wykonania.

Eksplorację danych (ang. *data mining*) możemy podzielić na dwie główne kategorie: eksplorację predykcyjną oraz odkrywanie wiedzy. Zagadnienie odkrywania wiedzy polega na znajdowaniu zależności pomiędzy danymi w warunkach, gdy posiadane dane są zbyt ubogie, aby je użyć do rozwiązania problemu predykcyjnego. Problemy predykcyjne to prognozowanie. Wyróżniamy dwa rodzaje problemów predykcyjnych: problemy regresyjne i problemy klasyfikacyjne. Problemy regresyjne (inaczej ilościowe) to takie, w których odpowiedź jest wartością liczbową, na przykład prognozowanie zysku lub strat firmy, prognoza produkcji. Problem klasyfikacji (inaczej jakościowy) to taki, w którym na wyjściu otrzymujemy przydział do danej klasy, np. niskie lub wysokie ryzyko straconego kredytu.

Obecnie najważniejszym medium wymiany informacji jest sieć Internet. Mimo wielu niezaprzeczalnych zalet Internetu jako medium komunikacyjnego, łatwo również wymienić jego wady, które są źródłem licznych zagrożeń. Tarczą, która ma chronić nas przed tymi zagrożeniami, są miedzy innymi systemy wykrywania intruzów. Zadaniem takich systemów jest wykrywanie ataków na środowisko serwerów i serwisów biznesowych i na środowisko sieciowe.

Celem niniejszego artykułu było dokonanie porównania różnych metod eksploracji danych pod kątem skuteczności w wykrywaniu ataków na infrastrukturę sieciową systemów informatycznych. Wykrycie ataku to problem klasyfikacji ruchu sieciowego na ruch bezpieczny albo ruch towarzyszący atakom na infrastrukturę lub nadużyciom w biznesowych systemach IT.

Do przeprowadzenia badań wykorzystano pakiet STATISTICA 10 PL. Liczne artykuły, opisujące modele i metody zaimplementowane w tym pakiecie, można znaleźć w czytelni na portalu firmy StatSoft [1]. W pracy [2] opisano, jak wykorzystać program STATISTICA oraz metody eksploracji danych do wykrywania nadużyć, na przykładzie transakcji wykonanych w sklepie internetowym (w tym przypadku chodzi o wykrycie klientów, którzy nie zapłacą za towar zakupiony w sklepie internetowym). Dane o transakcjach w sklepie internetowym pochodzą z konkursu DM Cup 2005 i można je znaleźć na portalu [3]. Praca [4] traktuje o metodach, jakie są stosowane w eksploracji danych. Natomiast [5] opisuje problem − predykcyjnej eksploracji danych, podobny do tego, jaki jest przedmiotem niniejszego artykułu. W [5] pokazano również przykład tworzenia odpowiedniego projektu eksploracji danych w programie STATISTICA. Artykuł [6] opisuje proces tworzenia projektu data mining w STATISTICA. Praca [7] opisuje metody wykrywania nadużyć, na przykładzie wykrywania prania brudnych pieniędzy. Wymienione powyżej publikacje były znaczące na etapie studialnym niniejszych badań.

2. Analiza danych

Do badań zostały użyte dane z konkursu KDD Mining Cup 1999 [8]. Zadaniem tego konkursu było opracowanie klasyfikatora zdolnego do rozróżniania połączeń sieciowych z przeznaczeniem do systemu mającego odróżniać normalne połączenie od ataku lub włamania. Zbiór danych obejmuje różnego rodzaju ataki symulowane na wojskowym środowisku sieciowym. Ataki dzielą się na cztery główne grupy: Denial of Service (DoS), Remote to Local (R2L), User to Root (U2R) oraz Probe.

Zbiór uczący składa się z 4 898 431 rekordów i zawiera 24 rodzaje ataków. Zbiór testowy jest złożony z 2 984 154 rekordów i dodatkowo zawiera 14 ataków, których nie było w zbiorze uczącym.

Każdy rekord opisany jest 24 zmiennymi [9]:
(1) *duration* – czas połączenia w sekundach, (2) *protocol_type* – rodzaj protokołu, (3) *service* – rodzaj usługi, (4) *flag* – połączenie normalne lub błędne, (5) *src_bytes* – liczba bajtów przesłanych od źródła do celu, (6) *dst_bytes* – liczba bajtów przesłanych od celu do źródła, (7) *land* – 1 jeżeli połączenie jest z tego samego hosta lub portu, 0 w przeciwnym wypadku, (8) *wrong_fragment* – liczba błędnych fragmentów, (9) *urgens* – liczba pakietów z flagą „*urgent*”, (10) *hot* – liczba wskaźników “*hot*”, (11) *num_failed_logins* – liczba logowań zakończonych niepowodzeniem, (12) *logged_in* – *1* jeżeli użytkownik zalogowany, *0* w przeciwnym wypadku, (13) *num_compromised* – liczba skompromitowanych warunków, (14) *root_shell* – *1* jeżeli uzyskano dostęp do powłoki Root, *0* w przeciwnym wypadku, (15) *su_attempted* – *1* jeżeli użyto komendy “*su root*”, *0* w przeciwnym wypadku, (16) *num_root* liczba dostępów do konta “*Root*”, (17) *num_file_creations* – liczba operacji tworzenia pliku, (18) *num_shells* – liczba monitów powłoki, (19) *num_access_files* – liczba operacji kontroli dostępu na plikach, (20) *num_outbound_cmds* – liczba komend wychodzących w sesji *ftp*, (21) *is_host_login* – *1* jeżeli zalogowany na konto “*hot*”, *0* w przeciwnym wypadku, (22) *is_guest_login* – *1* jeżeli zalogowano na koncie gościa, *0* w przeciwnym wypadku, (23) *count* – liczba połączeń do tego samego hosta w przeciągu ostatnich 2 sekund, (24) *class* – klasa ruchu.

Następnie uwzględniono 5 zmiennych dotyczących tego samego hosta:
(25) *serror_rate* – procent połączeń z błędem “*SYN*”, (26) *rerror_rate* – procent połączeń z błędem “*REJ*”, (27) *same_srv_rate* – procent połączeń do tego samego serwisu, (28) *diff_srv_rate* – procent połączeń do innego serwisu, (29) *srv_count* – liczba połączeń do tej samej usługi w przeciągu ostatnich 2 sekund.

Kolejnych 13 zmiennych dotyczy samej usługi:
(30) *srv_serror_rate* - procent połączeń z błędem “*SYN*”, (31) *srv_rerror_rate* - procent połączeń z błędem “*REJ*”, (32) *srv_diff_host_rate* – procent połączeń do innego hosta, (33) *dst_host_rate* – procent połączeń do tego samego hosta docelowego, (34) *dst_host_count* –

liczba połączeń do tego samego hosta docelowego, (35) *dst_host_srv_count* – liczba połączeń do tego samego hosta docelowego i tej samej usługi, (36) *dst_host_same_srv_rate* – procent połączeń do tego samego hosta docelowego i do tej samej usługi, (37) *dst_host_diff_srv_rate* – procent różnych usług połączonych do danego hosta, (38) *dst_host_same_src_port_rate* – procent połączeń do hosta z tym samym portem źródłowym, (39) *dst_host_srv_diff_host_rate* – procent połączeń do tej samej usługi od różnych hostów, (40) *dst_host_serror_rate* – procent błędnych połączeń do hosta z błędem „*SYN*”, (41) *dst_host_srv_serror_rate* – procent błędnych połączeń do hosta I tej samej usługi z błedem „*SYN*”, (42) *dst_host_rerror_rate* – procent błędnych połączeń do hosta z błędem „*REJ*”.

Autorom nie udało się znaleźć prac ze wskazaniem explicite metod, jakich używali laureaci konkursu. Należy uznać, że brak jest takiej publikacji. Jedyną znalezioną informacją była notatka na temat drużyny, która zajęła trzecie miejsce [10], w której podano, że drużyna korzystała z algorytmu „Fragment”, autorstwa V. Miheev i V. Pereversev-Orlov. Udało się znaleźć też kilka artykułów opisujących pracę nad tym zbiorem danych. Autorzy pracy [11], korzystają z metod J48, Bayesa, drzew NB, lasu losowego, drzew losowych, sieci neuronowych i SVM. Prace [12] i [13] opisują zastosowanie metody SVM. Autorzy pracy [14] opisują wykorzystanie sieci neuronowych do znalezienia klasyfikatora ruchu sieciowego, wykrywającego ataki na infrastrukturę sieciową.

3. Opis metod eksploracji danych

W projekcie eksploracji danych zostaną użyte następujące metody:

1. K-najbliższych sąsiadów

Metoda k-najbliższych sąsiadów (ang. *k Nearest Neighbours – kNN*) służy do rozwiązywania problemu klasyfikacji. Zasada działania jest dosyć prosta. Chcąc sklasyfikować obiekt porównujemy jego wektor cech z wektorami cech obiektów zbioru uczącego. Ze zbioru uczącego wybieramy k najbliższych sąsiadów. Odległość porównujemy wcześniej zadaną metryką. Obiekt przydzielamy do klasy, która najliczniej występuje w zbiorze k najbliższych sąsiadów.

2. Drzewa klasyfikacyjne i regresyjne

Drzewa klasyfikacyjne i regresyjne (ang. *Classification and Regression Trees – C&RT*), jak wskazuje nazwa, służą do rozwiązywania problemów regresyjnych oraz klasyfikacyjnych. Metodę opisano dokładnie w pracach [15] i [17]. Algorytm buduje drzewo na podstawie logicznych warunków podziału typu IF THEN. W metodzie C&RT możemy wyróżnić cztery główne etapy:

a) budowanie drzewa – poprzez rekurencyjny podział węzłów. Każdy węzeł przypisywany jest do danej klasy na podstawie podziału klasy w zbiorze uczącym i macierzy kosztów,
b) zatrzymanie budowy drzewa – po tym etapie mamy „maksymalne" drzewo, jakie mogło zostać wybudowane. Takie drzewo najprawdopodobniej zawiera informacje nadmiarowe,
c) przycinanie drzewa – zsunięcie z drzewa „maksymalnego" nadmiarowych gałęzi,
d) dobór drzewa – przywrócenie niektórych gałęzi usuniętych w kroku trzecim, aby zwiększyć skuteczność metody.

3. Drzewa CHAID

Metoda CHi-squared Automatic Interaction Detection (CHAID) służy do rozwiązywania jakościowych oraz ilościowych problemów predykcyjnej eksploracji danych. Metodę i model omówiono tu na podstawie prac [15] i [16]. Algorytm CHAID buduje drzewo, z którego węzłów mogą wychodzić więcej niż dwie gałęzie. Metoda wybiera ze zbioru zmiennych te, które mają największy wpływ na zmienną przewidywaną. Do wyznaczania kolejnych podziałów wykorzystywany jest test Chi-kwadrat (dla zmiennych jakościowych i problemów klasyfikacyjnych) lub test F Fishera (dla zmiennych ilościowych i problemów regresyjnych). Przebieg działania algorytmu składa się z następujących kroków:

a) przygotowanie predykatorów – podział predykatorów ilościowych na klasy jakościowe. Jeżeli predykatory są jakościowe, klasy są już utworzone,
b) łączenie kategorii – dla każdych dwóch kategorii każdego predykatora liczony jest test Chi-kawdrat (dla zmiennych jakościowych) lub test F (dla zmiennych ilościowych). Wartość testu jest oceną zależności dwóch kategorii. Jeżeli ocena jest na poziomie p, kategorie są łączone i krok jest powtarzany. W przeciwnym wypadku obliczana jest poprawka Bonferrioniego p-value dla predykatora,
c) wybór zmiennej do dzielenia – wybierany jest predykator o najmniejszej wartości p-value. Jest to predykator, który daje najbardziej istotny podział. Jeśli wartość poziomu p dla każdego predykatora jest niższy niż poziom p dla podziału, to dalsze podziały nie są wykonywane i węzeł jest liściem drzewa. Proces jest powtarzany do momentu, gdy możliwe są dalsze podziały.

Wyróżniamy dwie odmiany algorytmu: podstawowy algorytm CHAID i wyczerpujący algorytm CHAID. Algorytm wyczerpujący różni się od podstawowego tym, że wersja wyczerpująca łączy zmienne ilościowe aż do momentu uzyskania dwóch kategorii dla każdego predykatora.

4. Sieci neuronowe

Sieci neuronowe mają bardzo szerokie zastosowanie. Można je stosować do rozwiązywania problemu klasyfikacji. Opis sieci neuronowych można znaleźć w wielu źródłach, na przykład w [18]. Sieć neuronowa jest to zbiór połączonych ze sobą elementów nazywanych neuronami. Sama idea neuronu i sieci neuronowej została zainspirowana

biologiczną budową mózgu. Jednak sztuczny neuron jest tworem silnie uproszczonym w porównaniu do neuronu rzeczywistego. Neuron ma kilka wejść. Do każdego wejścia przypisana jest odpowiednia waga. Przez każde wejście do neuronu może być przekazywany sygnał, który jest przemnażany przez wagę danego wejścia. Jeśli suma sygnałów jest większa niż zadana wartość progowa, neuron jest pobudzany i sam wysyła sygnał przez swoje wyjście do następnego neuronu. Z neuronów możemy tworzyć sieci neuronowe o różnych architekturach. Architektura sieci zależy od jej zastosowania. Najczęściej stosowane są sieci wielowarstwowe. Składają się one z kilku warstw neuronów. Wyjście każdego neuronu danej warstwy połączone jest z wejściem każdego neuronu warstwy następnej. W takiej sieci wyróżniamy warstwę wejściową, wyjściową oraz warstwy ukryte. Do neuronów warstwy wejściowej podawane są odpowiednio przetworzone sygnały, które cechują stan badanego obiektu. Następnie te sygnały przetwarzane są przez warstwy ukryte i w końcu trafiają do warstwy wyjściowej, która podaje rozwiązanie danego problemu. Jednak, aby sieć neuronowa działała poprawnie, musi przejść proces uczenia. Aby nauczyć sieć, musimy mieć tak zwany zbiór uczący. Jest to zbiór krotek zawierających informacje, jakie wyjście ma być pobudzone dla danych wartości wejściowych. Następnie każdą wartość wejściową podajemy na wejście sieci i sprawdzamy, jaki neuron wyjściowy został pobudzony. Jeżeli został pobudzony inny neuron wyjściowy niż odpowiadający danemu wejściu, należy zmodyfikować wagi wejść neuronów tak, aby wynik był jak najbliższy oczekiwanemu. Procedurę powtarzamy dla wszystkich krotek ze zbioru uczącego. Możemy też wyróżnić pojęcie zbioru testowego. Do przetestowania działania sieci po zakończeniu procesu uczenia służy zbiór testowy. Jest to część zbioru uczącego, używana do testowania nauczonej sieci.

5. Funkcje sklejane MARSplines

Wielozmienna regresja adaptacyjna z użyciem funkcji sklejanych (ang. *Multivariate Adaptive Regression Splines – MARSplines*) to metoda do rozwiązywania problemów regresyjnych oraz klasyfikacyjnych. Opisano ją dość szczegółowo w publikacjach na portalu StatSoft [15]. Metoda ta jest procedurą nieparametryczną, co oznacza, że nie wymaga założeń na temat funkcyjnych zależności między zmiennymi. W metodzie tej stosuje się strategię „dziel i rządź”. Przestrzeń wejściowa jest dzielona części i dla każdej części używana jest inna funkcja regresji. Model metody MARSplines możemy przedstawić w postaci wzoru (1):

$$y = \beta_0 + \sum_{m=1}^{M} \beta_m h_m(X) \tag{1}$$

gdzie: y – zmienna objaśniana,

m=1,…,M – funkcyjne składniki modelu,

$\beta_0, \beta m$ – stałe modelu, rzędna początkowa i rzędna ważona wagami β_m suma jednej lub wielu zmiennych bazowych $h_m(X)$,

h_m – funkcja bazowa,

X – zbiór wszystkich predykatorów, zbiór zmiennych predykcyjnych i ich interakcji.

Funkcje bazowe wraz z parametrami znajdywane są za pomocą metody najmniejszych kwadratów. W procesie tworzenia modelu możemy wyróżnić trzy główne etapy:

a) budowa prostego modelu – budowany jest prosty model ze stałą funkcją bazową,
b) rozwój modelu – do modelu dodawane są kolejne, coraz bardziej złożone funkcje, które poprawiają działanie modelu,
c) czyszczenie modelu – z modelu usuwane są funkcje, których znaczenie jest najmniej istotne.

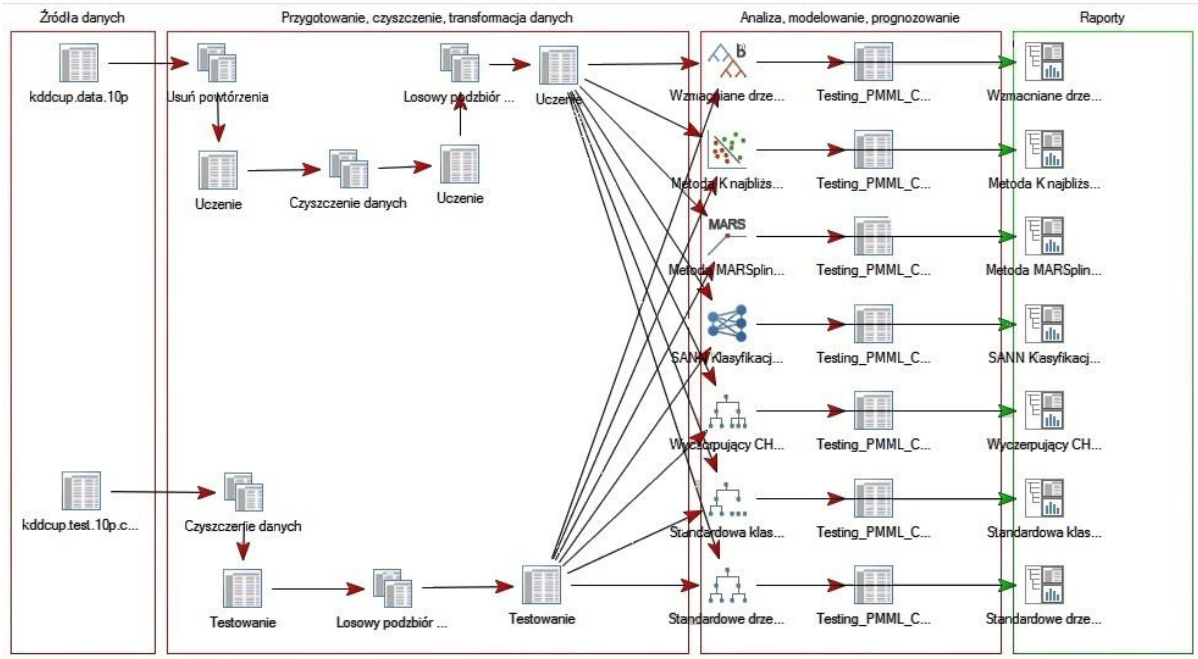
6. Drzewa wzmacniane

Metoda drzewa wzmacnianego służy do rozwiązywania problemów regresyjnych i klasyfikacyjnych. Jest to jedna z najpopularniejszych metod predykcyjnej eksploracji danych. Metodę tą również opisano dość szczegółowo w publikacjach na portalu StatSoft [15]. Tworzy się w niej ciąg prostych drzew binarnych tak, aby każde kolejne drzewo służyło do predykcji reszt drzewa poprzedniego. W efekcie metoda daje bardzo dobry model predykcyjny, nawet, gdy zbiór uczący nie jest liczny lub gdy zależności pomiędzy zmiennymi obserwowanymi są bardzo złożone. Aby ograniczyć wpływ problemu „przeuczenia" modelu, każde kolejne drzewo budowane jest na podstawie próby wylosowanej z ciągu uczącego.

4. Opis projektu eksploracji danych

Dla celów omawianych tu badań projekt eksploracji danych wykonano w środowisku STATISTICA 10 PL. Wyróżniamy w nim cztery główne części: (1) zdefiniowanie źródła danych, (2) przygotowanie, czyszczenie i transformacja danych, (3) analiza, modelowanie i prognozowanie, (4) tworzenie raportów.

W każdej części znajdują się węzły, które są odpowiednio ze sobą połączone. Połączenia symbolizują przepływ danych. Węzły mogą reprezentować arkusz danych, raport lub procedurę.



Rys. 1. Zrealizowany w programie STATISTICA 10 PL projekt eksploracji danych
Fig. 1. Data mining project carried out in STATISTICA 10 PL
Źródło: opracowanie własne.

Na rysunku 1 przedstawiono zdefiniowaną przestrzeń projektową. W części źródła danych widać dwa węzły. Są to arkusze danych *kddcup.data.10p* i *kddcup.test.10p.corrected*. Zawierają one po 10% danych ze zbiorów uczącego i testowego przygotowanego na konkurs KDD CUP 1999 i udostępnione uczestnikom przez organizatorów. Dane z arkusza *kddcup.data.10p* połączone są z węzłem *usuń powtórzenia*, który usuwa z arkusza powtarzające się rekordy. Ten zabieg ma zminimalizować ryzyko wystąpienia problemu, w którym model nauczy się dobrze rozpoznawać jedynie obiekt najliczniej występujący w zbiorze uczącym. Arkusz na wyjściu ma nazwę *Uczenie*, ponieważ zostanie on użyty jako zbiór uczący. Następnie dane trafiają do węzła *Czyszczenie danych*. Usuwa on z arkusza rekordy niekompletne. Następnie dane trafiają do węzła *Losowy podzbiór przypadków*, który losuje do zbioru uczącego 20 tys. rekordów. Wylosowane rekordy trafiają do zbioru uczącego. Dane z *kddcup.test10.corrected*, które posłużą jako zbiór testowy przechodzą przez węzły *Czyszczenie danych* i *Losowy podzbiór przypadków*. Jako zbiór testowy losuje się 5000 rekordów. Dane z arkuszy Uczenie i Testowanie trafiają do węzłów tworzących modele do klasyfikacji: (I) Wzmacniane drzewa klasyfikacyjne z wdrożeniem, (II) Metoda k-najbliższych sasiadów z wdrożeniem (klasyfikacja), (III) *Metoda MARSplines z wdrożeniem dla zadań klasyfikacji*, (IV) *SANN Klasyfikacja z wdrożeniem*, (V) *Standardowa klasyfikacja CHAID z wdrożeniem*, ((VI) *Wyczerpujący CHAID dla klasyfikacji z wdrożeniem*, (VII) *Standardowe drzewa klasyfikacyjne (CRT) z wdrożeniem*.

Na wyjściu otrzymujemy raporty z działania modelu i arkusze z wynikiem klasyfikacji zbioru testowego. Do klasyfikacji użyto 20 zmiennych, wybranych za pomocą włączonego aparatu automatycznego dobru zmiennych.

4. Analiza wyników badań

W ramach badań wykonano trzykrotnie modele dla każdej z metod. Zbiór uczący miał rozmiar 20000 rekordów i był wybierany losowo, natomiast zbiór testowy 5000 rekordów.

Tabela 1

Wyniki badań

Metoda eksploracji	Klasyfikacja		Atak	
	Średnia	Odchylenie standardowe	Średnia	Odchylenie standardowe
K-najbliższych sąsiadów	71,84%	0,81%	72,78%	0,74%
Wzmacniane drzewa klasyfikacyjne	8,16%	5,18%	83,34%	7,24%
MARSplines	60,29%	29,98%	66,34%	34,37%
Sieci neuronowe	20,42%	8,50%	72,05%	8,38%
Standardowy CHAID	19,92%	0,31%	20,05%	0,18%
Wzmacniane CHAID	19,91%	0,30%	20,05%	0,18%
Drzewa CRT	21,04%	9,21%	61,25%	27,21%

Źródło: opracowanie własne

W tabeli 1 kolumna *Klasyfikacja* opisuje, jaki procent obiektów został sklasyfikowany prawidłowo, tzn., ile procent rekordów zostało trafnie rozpoznanych jako ruch sieciowy normalny, a ile zostało rozpoznanych jako konkretny rodzaj ataku. Kolumna *Atak* zawiera procentowe oszacowanie liczby obiektów prawidłowo sklasyfikowanych, jednak w tym wypadku brano pod uwagę tylko klasyfikację na ruch sieciowy normalny oraz na ruch wskazujący na atak, bez rozróżnienia rodzaju ataku.

W ponad 70% przypadków najskuteczniejsza dla *Klasyfikacji* była metoda kNN ze średnim wynikiem trafności klasyfikacji i z niskim poziomem odchylenia standardowego. Pozostałe metody nie okazały się zbyt skuteczne.

Dla *Ataku* najskuteczniejsza była metoda drzew wzmacnianych z wynikiem wykrywania ataku w ponad 80%. Sieci neuronowe i metoda k-najbliższych sąsiadów uzyskały wynik ponad 70%. Sieci neuronowe otrzymały odchylenie standardowe na poziomie 8%, zatem stosunkowo bardzo duże.

5. Podsumowanie

Dla *Klasyfikacji* do przyjęcia jest tylko metoda kNN. Mimo swojej prostoty udało jej się również uzyskać bardzo dobry wynika dla *Atak*. Dla trafności wykrywania ataku *Atak* w miarę skuteczne okazały się metody: drzew wzmacnianych i sieci neuronowych. Jednak w przypadku tych dwóch metod odchylenie standardowe było na wysokim poziomie i wyniosło 7%. Wyniki przeprowadzonych badań pokazują, że warto stosować eksplorację danych do wykrywania nadużyć. Aby jeszcze poprawić wyniki, należałoby się zastanowić, w jaki sposób lepiej generować zbiór uczący.

Bibliografia

1. StatSoft Polska, Czytelnia StatSoft. http://www.statsoft.pl/czytelnia, 20/05/2012.
2. Demski T.: Tworzenie i stosowanie modelu data mining za pomocą Przepisów STATISTICA Data Miner na przykładzie wykrywania nadużyć. StatSoft Polska, http://www.statsoft.pl/czytelnia/czytelnia.html, 20/05/2012.
3. DM Cup.: Data mining cup, "Prudsys", http://www.data-mining-cup.de/, 18/04/2012.
4. Sokołowski A.: Metody stosowane w data mining, StatSoft Polska, 2002.
5. Wątroba J.: Przykłady rozwiązania zagadnienie predykcyjnego za pomocą technik data mining. StatSoft Polska, 2002.
6. Wątroba J., Kowalski T., Demski T.: Data mining i jego realizacja w STATISTICA data miner. StatSoft Polska, 2002.
7. Kuijlen T., Migut G.: Wykrywanie nadużyc i prania brudnych pieniedzy. StatSoft Polska, 2004.
8. KDD Cup: ACM Special Interest Group on Knowledge Discovery and Data Mining. http://www.sigkdd.org/, 18/04/2012.
9. Stolfo S., Wei F., Lee W.: Promodromidis A.: Cost-based Modeling and Evaluation for Data Mining With Application to Fraud and Intrusion Detection: Results from the JAM Project. Columbia University, New York 1999.
10. Miheev V., Vopilov A., Shabalin I.: The MP13 Approach to the KDD'99 Classifier Learning Contest. ACM SIGKDD, Moskwa 2000.
11. Tavallaee M., Bagheri E., Lu W., Ghorbani A.: A Detailed Analysis of the KDD CUP 99 Data Set. Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defence Applications, 2009.
12. Fortuna C., Fortuna B., Mohorčič M.: Anomaly detection in computer networks using linear SVMs. Slovenian Research Agency, 2002.

13. Eskin I.: Multi-Class SVMs for Intrusion Detection, http://www.math.tau.ac.il/~mansour/ml-course-10/students_projects/Final_Project_-_Intrusion_Detection.pdf, 20/05/2012.
14. Saima Munawar M.N.H.A.B.: Anomaly Detection through NN Hybrid Learning with Data Transformation Analysis. International Journal of Scientific & Engineering Research, tom 3, nr 1, 2012.
15. StatSoft Polska: Elektroniczny Podręcznik Statystyki, StatSoft Polska, 2006, 20/05/2012.
16. Nowak-Brzezińska A.: Drzewa klasyfikacyjne, http://informatyka.us.edu.pl/drzewa-klasyfikacyjne/, 11/10/2010.
17. Lewis R.J.: An Introduction to Classification and Regression Tree (CART) Analysis. Harbor-UCLA Medical Center, California 2000.
18. Kwaśnicka H., Markowska-Kaczmar U.: Sieci neuronowe w zastosowaniach. Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław 2005.

Abstract

This article describes how to use data mining methods to detect fraud. Introduction provides a brief introduction to the concept of data mining. Then the data are described that were used in the study. The study used data from the KDD Cup 99th The data include simulated attacks on a military network environment. Then methods data mining, such as k-nearest neighbor, wood-reinforced, neural networks, trees, CHAID, C & RT tree, MARSplines have been described. These methods have been used to create a data mining project in the program STATISTICA PL 10.