Penyesuaian Proses *Generating Cluster* Secara Periodik Pada Algoritma Denstream Untuk Analisis Sistem Deteksi Anomali Trafik

Adustment Process Generating Cluster Periodically on Denstream Algorithm for Analysis Traffic Anomaly Detection

Nana Permana[1], Yudha Purwanto[2], Fiky Yosef Suratman[3]

[1,2,3]Prodi S1 Teknik Komputer, Fakultas Teknik Elektro, Universitas Telkom

[1]nanapermana@student.telkomuniversity.ac.id, [2]omyudha@telkomuniversity.ac.id, fysuratman@telkomuniversity.ac.id

Abstrak

Dalam beberapa tahun terakhir internet mengalami perkembangan yang pesat Oleh karena itu terjadi trafik yang tinggi di dalam suatu jaringan komputer. Trafik yang tinggi dapat disebabkan oleh suatu serangan Jenis serangan yang umum dilakukan adalah *Denial of Service* (DoS) dan *Distributed Denial of Service* (DDoS). Oleh karena itu penting membangun sistem yang dapat mendeteksi serangan DDoS. Pada penelitian ini dibangun sebuah metode *Intrusion Detection System* (IDS) menggunakan algoritma Denstream modifikasi . Modifikasi pada proses *generating cluster* dilakukan secara periodik disesuaikan dengan karakteristik *datastream.* Hasil dari penelitian ini yaitu, sistem yang dibagun dapat bekerja dengan baik dalam deteksi dan membedakan antara *traffic* normal dan *traffic anomaly* dengan sedikit kesalahan deteksi pada *cluster* normal. Pada penelitian ini digunakan *purity* sebagai parameter uji untuk algortima Denstream modifikasi dimana menghailkan rataan *purity* sebesar 98.02% pada seluruh dataset yang digunakan.

Kata Kunci : ***Anomaly Traffic*, DDoS, Denstream, *Clustering***

Abstract

In recent years the Internet experience rapid growth because of that occurs high traffic in a computer network. High traffic can be caused by an attack this type of attack is common Denial of Service (DoS) and Distributed Denial of Service (DdoS). It is considered to build systems that can detect DDoS attacks. In this research we constructed a method Intrusion Detection System (IDS) using algorithm Denstream modified. Modifications in the process of generating the cluster is done periodically adjusted to the characteristics of the datastream. Results of this research is the systems can work well in the detection and distinguish between normal traffic and traffic anomalies with less error detection in normal *cluster*. In this research we use purity as parameter to rate our Denstream modified algorithm it produce average purity of 98.02% in all dataset used.

Keywords: traffic anomalies, DDoS, Denstream, *Cluster*ing

1. Pendahuluan

Distributed Denial of Service (DDoS) adalah suatu jenis serangan terhadap sebuah komputer atau server dengan salah satu cara membanjiri lalu lintas jaringan dengan banyak permintaan (*request flooding*) sehingga tidak dapat diakses oleh *user* yang berhak. Dalam penelitian ini kami akan menggunakan algoritma pengembangan dari *density-based* DBSCAN yang disebut algoritma Denstream Algoritma Denstream sendiri menggunakan teknik *micro-cluster* yang digunakan untuk mengindentifikasi *cluster* beserta atribut didalamnya dalam sebuah *data stream.*Pada tugas akhir ini kami akan menggunkan algoritma Denstream dan menggunkan teknik pengukuran jarak *Euclidean Distance*. Fokus penelitian tugas akhir ini menerapkan algoritma Denstream modifikasi pada proses generating *cluster* ke dalam sistem deteksi *anomaly traffic*.

2. Dasar Teori dan Perancangan

2.1. Sistem Deteksi Anomaly

Pada Deteksi anomali trafik dikenal istilah Intrussion Detection System (IDS) dan Intrussion Prevention System (IPS). Pada IDS sistem harus dapat memonitor dan mendeteksi suatu serangan yang terjadi pada jaringan. Teknik IDS ini digunakan untuk memonitor aktivitas jaringan dalam waktu tertentu dan juga menetapkan suatu nilai (threshold) sebagai parameter untuk mendeteksi suatu serangan. Pengembangan dari teknik IDS ini menjadi suatu acuan untuk adanya suatu teknik untuk mencegah serangan tersebut atau dikenal

sibagai Intrusion Prevent System (IPS) yang memiliki kemampuan dalam memantau dan mendeteksi serangan layaknya IDS kemudian melakukan aksi dalam mengatasi serangan secara otomatis.

2.2. Distributed-Denial of Service (DDoS)

Denial of Service (DoS) adalah salah satu jenis serangan menggunakan *request* yang sah untuk membanjiri lalu lintas host target menyebabkan host target menjadi *hang*, *crash*, atau *reboot* [1]. Seluruh *resource* yang dimiliki target digunakan untuk melayani serangan sehingga *resource* yang dimiliki target menjadi habis. DDoS merupakan bentuk serangan DoS yang dilakukan secara terdistribusi oleh banyak *zombie* yang dikendalikan oleh *botnet*, dimana *botnet* [2] sendiri dikendalikan oleh *botmaster*. Inilah yang membuat serangan DDoS lebih sulit ditelusuri keberadaan penyerang sebenarnya. Dalam definisi secara luas yaitu setiap upaya untuk merusak dan menolak segala layanan yang ada sasaran serangan dalam DoS/DDoS adalah *bandwidth* untuk membuat sumber daya *bandwidth* penuh atau tidak tersedia lagi bagi user yang akan mengakses. akibat dari hal tersebut sumber daya komputasi baik dari proses, *memory*, *buffer* pada *server* maupun *node* jaringan [3] akhirnya menjadi *crash/down* sehingga tidak dapat melayani servis yang diminta *user*.

2.3. Density Based *Cluster*ing

Clustering merupakan suatu metode dalam data mining yang dapat digunakan untuk mendeteksi suatu anomali trafik. *Clustering* merupakan teknik untuk memisahkan data yang tidak diketahui informasinya kedalam struktur kelompok yang memiliki kemiripan maksimum dengan data lain dalam satu *cluster*. *Density based Clustering* merupakan suatu teknik dalam *clustering* dimana *cluster* terbentuk pada bagian yang memiliki kerapatan tertentu [4].

2.4. Denstream *Cluster*ing

*Cluster*ing [5] Salah satu cara mendeteksi anomali trafik adalah dengan metode *clustering* banyak algoritma *clustering* yang ada akan tetapi kurang efektif penggunaannya dalam deteksi anomali trafik maka digunakanlah algoritma Denstream. Algoritma Denstream jika digunakan untuk melakukan *clustering* pada datastream maka akan membutuhkan proses *Damped window* [6] dimana metode *windowing* ini akan mengurangi beban komputasi yang dilakukan karena berat data point akan selalu berkurang terhadap fungsi waktu. Algoritma Denstream menggunakan teknik *micro-cluster* yang digunakan untuk memproses banyak *outlier* dalam sebuah *data stream*, *micro-cluster* pada algoritma *Denstream* dibedakan menjadi tiga yaitu *Core micro-cluster(c-micro-cluster)* , *Potensial C-micro-cluster(p-micro-cluster)* , dan *Outlier C-micro-cluster(o-micro-cluster)*. *c-micro-cluster* merupakan micro-*cluster* utama pada algoritma ini dan *c-micro-cluter* mempunyai tiga atribut utama yaitu *weight*, *center* dan *radius*, ketiga atribut tersebut dapat dikalkulasikan untuk *set point* $p_1, p_2, p_3, \ldots p_n$ pada waktu t dengan *time stamp* $T_1, T_2, T_3, \ldots T_n$ dengan formula berikut:

$$w = \sum_{i=1}^{n} f(t - T_i) \quad (1)$$

$$c = \frac{\sum_{i=1}^{n} f(t-T_1)p_i}{w} \quad (2)$$

$$r = \frac{\sum_{i=1}^{n} f(t-T_{ij})dist(p,c)}{w} \quad (3)$$

$$CF_1 = \sum_{j=1}^{n} f(t - t_{ij})P_{ij} \quad (4)$$

$$CF_2 = \sum_{j=1}^{n} f(t - t_{ij}){P_{ij}}^2 \quad (5)$$

p-micro-cluster merupakan *micro-cluster* yang berpotensi menjadi *cluster* utama maupun menjadi sebuah outlier yang akan dihapus tergantung pada berat dari *p-micro-cluster* sendiri sedangkan *o-micro-cluster* merupakan *micro-cluster* [7]yang berpotensi menjadi p-micro-*cluster* maupun menjadi sebuah *outlier* tergantung pada berat dari *o-micro-cluster* tersebut. Algoritma Denstream selalu melakukan pertimbangan berat terhadap setiap micro *cluster*nya apakah micro *cluster* tersebut layak menjadi micro *cluster* utama atau hanya akan dihapus karena dianggap hanya sebagai outlier. Sebelum datangnya set point pada Denstream akan dilakukan perhitungan Tp dimana Tp adalah waktu minimum yang dibutuhkan untuk *p-micro-cluster* berubah menjadi *outlier*. Formula dari Tp adalah sebagai berikut:

$$Tp = \left[\frac{1}{\lambda}\log\left(\frac{\beta\mu}{\beta\mu - 1}\right)\right] \quad (6)$$

Denstream akan menerima set point dimana digunakan DBSCAN untuk membuat initial set dari *p-micro-cluster* . Saat initial *p-micro-cluster* terbentuk , Denstream menunggu sampai sebuah point datang dari *stream*. Point yang baru datang ini awalnya mencoba bergabung ke *p-micro-cluster* terdekat ketika point tersebut kurang dari radius *p-micro-cluster* terdekat maka dia akan bergabung dengan *p-micro-cluster* tersebut lalu radius,center dan berat dari *p-micro-cluster* akan dilakukan *update*

Apabila point gagal bergabung dengan *p-micro-cluster* , maka point akan mencoba bergabung dengan *o-micro-cluster*. Ketika *o-micro-cluster* yang baru mempunyai berat yang cukup maka *o-micro-cluster* akan menjadi *p-micro-cluster* dan dihapus dari *outlier buffer*. Ketika point tidak bisa bergabung dengan *micro-cluster* lainnya maka Denstream membuat *o-micro-cluster* pada point tersebut dan menempatkannya di *outlier buffer*. Setiap ada point yang datang Denstream secara periodik mengecek dan menghapus semua *o-micro-cluster* yang memiliki berat kurang dari *lower limit of weight*. *Lower limit of weight* didefinisikan sebagai berikut.

$$\xi(t_c, t_o) = \frac{2^{-\lambda(t_c - t_o + T_p)} - 1}{2^{-\lambda T_p} - 1} \quad (7)$$

Dimana t_c adalah waktu sekarang dan t_o adalah waktu yang dibutuhkan untuk membuat *o-micro-cluster*. Selanjutnya dilakukan proses generating *cluster* dimana *cluster* yang sudah terbentuk akan dicari centernya dengan menggunakan rumus:

$$C = \frac{\sum_j^n f(t - t_{ij}) x_{ij}}{w_{total}} \quad (8)$$

kemudian center dari *cluster* tersebut dianggap sebagai data point baru yang kemudian dilakukan kembali proses algoritma dbscan.

2.5. Euclidean Distance

Euclidean Distance adalah metrika yang paling sering digunakan untuk menghitung kesamaan 2 vektor. Euclidean distance menghitung akar dari kuadrat perbedaan 2 vektor, rumus dari euclidian distance adalah:

$$d(p,q) = \sqrt{\sum_{i=1}^{n} (q_1 - p_1)^2} \quad (9)$$

Pada penelitian ini untuk menghitung jarak antar titik / data digunakan rumus euclidian distance tiga dimensi:

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + (p_3 - q_3)^2}\,. \quad (10)$$

fitur dari dataset dijadikan titik/point yang kemudian dihitung menggunakan euclidian distance.

2.6. Dataset DARPA 1998

Pada penelitian ini menggunakan *dataset* yang sudah sering digunakan pada peneletian serupa sebelumnya. Untuk pengujian *traffic* DDoS digunakan *dataset* DARPA 1998 [8] yang merupakan hasil datastream [9]data tersebut sudah menjalani proses *preprocessing* agar mudah dianalisia. Untuk pengujian metode IDS yang dirancang, dilakukan simulasi menggunakan bahasa pemrograman Java.

2.7. Parameter Uji

Pada penelitian ini, pengujian sistem deteksi anomali trafik ditentukan dengan menghitung performansi algoritma dalam melakukan proses deteksi dengan parameter purity pengaruh parameter algoritma terhadap hasil *cluster*ing, analisis hasil *cluster*ing menggunakan tabel konvolusi seperti pada table 1 :

Tabel 1 Konvolusi

Prediksi	*Aktual*	
	Normal	*Serangan*
Normal	Normal = Normal *(TN)*	Normal = Serangan *(FP)*
Serangan	Serangan =Normal*(FN)*	Serangan=Serangan *(TN)*

True positive (TP) adalah kondisi dimana algoritma mendeteksi data sebagai serangan dan kelanjutan sebenarnya memang data tersebut merupakan serangan. *True nagative* (TN) adalah dimana algoritma mendeteksi data sebagai kondisi normal dan kenyataannya memang data tersebut merupakan data normal. *False positive* (FP) adalah dimana kondisi algoritma mendeteksi data dengan kondisi normal tetapi disebut

sebuah serangan. *False negative* (FN) adalah kondisi dimana algoritma melakukan salah deteksi yang menyatakan data dengan kondisi serangan disebut segabagi kondisi normal.

Purity

Parameter uji yang digunakan pada penelitian kali ini adalah *purity* untuk pentuan keberhasilan dari algoritma Denstream modifikasi. *Purity* merupakan ukuran tingkat kemurnian dari suatu *cluster* dimana purity menghitung ada atau tidaknya suatu data yang berbeda di seluruh *cluster* dengan perhitungan sebagai berikut:

$$pur = \frac{\sum_{i=1}^{K} \left| \frac{C_i^d}{C_i} \right|}{K} X\ 100\% \quad (11)$$

Dimana $\left|C_i^d\right|$ merupakan jumlah data dengan label yang terbanyak lalu C_i merupakan total keseluruhan data yang ada di *cluster* tersebut kemudian K adalah total jumlah *cluster* yang terbentuk.

3. Pembahasan

Preprocessing adalah suatu proses untuk normalisasi sebuah data trafik agar mudah/cocok untuk digunakan pada proses pendeteksian. Pada penelitian sebelumnya menggunakan metode *preprocessing* memudahkan menganalisis dan meningkatkan hasil analisis yang dilakukan. Tujuan dalam proses *preprocessing* untuk melakukan mendapatkan fitur yang relevan dari *raw data,* dalam penelitian ini dilakukan pada dataset DARPA 1998 [8]. Fitur yang digunakan dapat dilihat pada Tabel 2.

Tabel 2 Ekstraksi Fitur

Nama Fitur	**Jenis Koneksi**	**Penjelasan**
Count	-	Jumlah *traffic* dalam satu *window*
IP_source	IP Source dan IP Destination sama	Jumlah *traffic* dari IP Source ke IP Destination yang sama
Protocol		Jumlah protocol yang sama
SYN		Jumlah *traffic* "SYN"
ACK		Jumlah *traffic* "ACK"
Port_Out		Jumlah *traffic* menuju ke port out yang sama
Length		Jumlah *traffic* dengan length yang sama
Different_Source	IP Destination sama	Jumlah *traffic* dengan IP Source berbeda
New_IP	-	Jumlah kemunculan IP baru

Penelitian Tugas Akhir ini adalah melakukan modifikasi pada proses *Generating Cluster* yaitu mengatur interval data secara periodik yang akan masuk pada setiap proses algoritma Denstream. Dianggap periode waktu = *t* maka selama periode *t* data akan masuk pada proses pertama Denstream yaitu proses DBSCAN setelah data sebanyak periode *t* selesai dilakukan proses DBSCAN maka data sebanyak periode *t* selanjutnya diproses oleh algoritma Denstream seperti dijelaskan pada Algoritma 1 dan Algoritma 2. Setelah selesai maka akan diambil center dari *cluster* yang sudah terbentuk untuk dijadikan data baru seperti dijelaskan pada Algoritma 3. Selanjutnya data yang berasal dari center tersebut akan digabungkan dengan data sebanyak periode *t* selanjutnya sampai data habis. Pada penelitian sebelumnya proses generating *cluster* dilakukan hanya jika diperlukan oleh peneliti. Dengan penggunaan interval data secara periodik diharapkan sistem menjadi lebih stabil.

Algoritma 1. Merging (DS, CS, Point, eps, β, μ)

1. Masukan point p ke p-micro*cluster* cp;
2. ***if*** rp (radius terbaru dari cp) ≤ eps then
3. Gabung p ke cp;

4. ***else***
5. Masukan point p ke o-micro*cluster* co;
6. ***if*** ro(radius terbaru dari co)≤ eps then
7. Gabung p ke co;
8. ***if*** w (weight baru dari co)> βμ then
9. Hapus co buat baru p-micro*cluster* dari co
10. ***end if***
11. ***else***
12. Buat o-micr*cluster* dari p
13. ***end if***
14. ***end If***
15. Simpan sebagai *Cluster* (CS1)

Algoritma 2. Denstream (DS, CS1, eps, β ,μ, λ)

1. ***get*** Tp=[1/λ log(βμ/(βμ-1))]
2. Dapatkan titik p selanjutnya dari waktu sekarang pada datastream;
3. Merging(p);
4. ***if*** (t mod tp) = 0 then
5. ***for*** setiap p-micro-*cluster* do
6. ***if*** wp (weight dari cp) < βμ then
7. Hapus cp;
8. ***end if***
9. ***end for***
10. ***for*** setiap o-micro-*cluster* Co do
11. ***get*** $\xi(t_c, t_o) = \frac{2^{-\lambda(t_c - t_o + T_p)} - 1}{2^{-\lambda T_p} - 1}$
12. ***if*** wo (weight dari co) < ξ then
13. Hapus co;
14. ***end if***
15. ***end for***
16. ***end if***
17. Simpan sebagai *Cluster* (CS2)

Algoritma 3. Generating *Cluster* (DS, CS2, Point)

1. ***get*** center dari *cluster*
2. Merge center dengan point p;
3. Do DBSCAN;
4. ***end:***
5. Simpan sebagai *Cluster* (CS)

3.1. Pengujian Dataset DDos

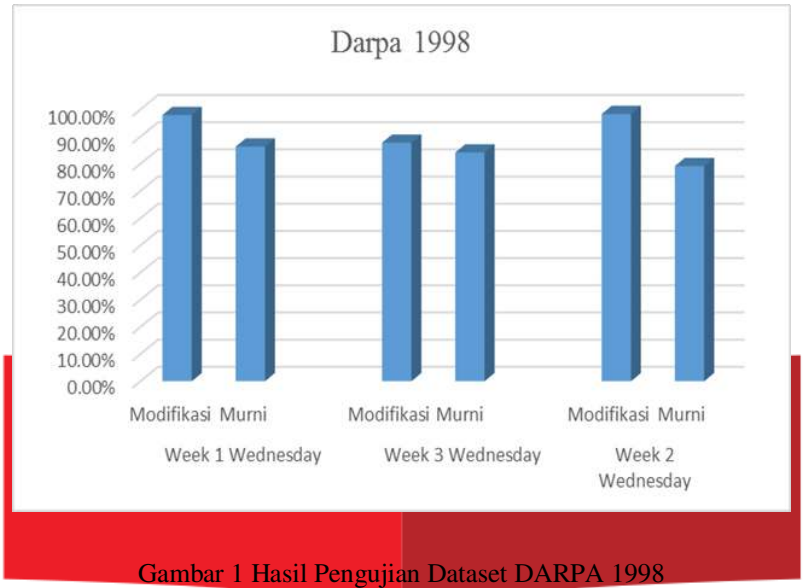
Proses *preprocessing* dahulu dilakukan pada dataset DARPA 1998, dikarenakan dataset darpa masih berupa *raw data.* Tujuan dari proses *preprocessing* untuk medapatkan karekteristik dari *traffic* DDoS sehingga hasil performansi deteksi yang dihasilkan lebih baik. Pengujian dilakukan pada *dataset* normal dan serangan mendapati hasil yang beragam. Hasil pengujian dari *dataset* DARPA 1998 dapat dilihat pada tabel 3 dan tabel 4.

Tabel 3. Hasil pengujian *week 1 wednesday*

Prediksi	Aktual			
	Normal	Smurf	Neptune	Back
Normal	33	29	-	-
Smurf	-	1950	-	-
Neptune	-	-	-	-
Back	-	-	-	-
Purity = 97.96 %				

Tabel 4. Hasil Pengujian *Week 2 Wednesday*

Prediksi	Aktual			
	Normal	Smurf	Neptune	Back
Normal	31`	12	-	-
Smurf	-	2002	-	-
Neptune	-	-	-	-
Back	-	-	-	-
Purity = 98.92 %				



Gambar 1 Hasil Pengujian Dataset DARPA 1998

Pada gambar 1 menggambarkan perbandingan *purity* antara algoritma Denstream dengan algoritma Denstream modifikasi pada proses *generating cluster*. Dimana algoritma Denstream modifikasi memiliki purity 97.96% sedangkan algoritma Denstream memiliki *purity* hanya 86.34% untuk dataset *DARPA week 1 Wednesday*. Untuk dataset *DARPA week 3 Wednesday* algoritma Denstream modifikasi unggul tipis dengan purity 87.74% sedangkan algoritma Denstream dengan *purity* 84.18%. untuk dataset *DARPA week 2 Wednesday* algoritma Denstream modifikasi juga unggul dengan *purity 98.34%*. Dengan menggunakan dataset DARPA 1998 terlihat bahwa Denstream modifikasi unggul dari segi parameter uji *purity* untuk ketiga dataset yang diuji. Hal ini disebabkan algoritma Denstream tidak menggunakan periode waktu dalam setiap proses algoritma didalamnya sehingga seluruh data akan langsung diproses dalam algoritma maka besar kemungkinan banyak data yang dihapus disebabkan penggunaan metode *Damped window* pada algoritma Denstream.

4. Kesimpulan

Kesimpulan dari penelitian ini, sistem dengan algoritma Denstream modifikasi pada proses generating *cluster* dapat membedakan antara *traffic* normal dan *traffic anomaly*. Pada saat sistem mendeteksi *cluster* serangan didalamnya tidak terdapat sama sekali data trafik normal tetapi saat deteksi *cluster* normal masih terdapat beberapa data trafik serangan didalamnya. Input parameter sangat menentukan bentuk dan jumlah *cluster* sehingga mempengaruhi juga purity dari *cluster* tersebut. Untuk pengembangan sistem deteksi *anomaly traffic* pada penenlitian selanjutnya, modifikasi sistem bisa dengan menggunakan datastream realtime untuk mendapatkan bentuk *cluster* secara langsung.

Daftar Pustaka

[1] Gou. Rui, G. Chang, R. hou, Y. Qin, B. Sun, A. Liu dan D. Peng, "Research on counter bandwidth depletion DDoS attacks based on Genetic," *International Conference on Natural Computation,* vol. III, 2007.

[2] Cao. Feng, M. Ester, W. Qian dan A. Zhou, "Density-Based *Cluster*ing over an Evolving Data Stream with Noise," *SIAM Conference Data Mining, Bethesda,* 2006.

[3] "Cyber System and Technology," Lincoln Laboratory Messachusetts Institute of Technology, 4 December 1998. [Online]. Available: http://www.ll.mit.edu/ideval/data/. [Diakses 23 October 2014].

[4] P. Danzig, J. Mogul, V. Paxson dan M. Schwartz, "WorldCup98," ACM SIGCOMM, [Online]. Available: http://ita.ee.lbl.gov/html/contrib/WorldCup.html.

[5] Wang. Huan, Y. Yu, Q. Wang dan Y. Wan, "A Density-Based *Cluster*ing Structure Mining Algorithm for Data Streams".

[6] Reddy. Rajani, R. Siva dan C. Malathi, "Techniques to Differentiate DDOS Attacks from Flash Crowd," *International Journal of Advanced Research in Computer Science and Software Engineering,* vol. 3, no. 6, pp. 295-299, June 2013.

[7] Purwanto Yudha, kuspriyanto dan Hendrawan, "Traffic Anomaly Detection in DDos Flooding Attack," *International Conference On Telecommunication System, Services, And Application,* vol. VIII, pp. 313-318, 2014.

[8] Matysiak. Martin, "Data Stream Mining," 2012.

[9] A. K. Mann dan N. Kaur, "Review Paper on *Cluster*ing Techniques," *Global Journal of Computer Science and Technology,* vol. XIII, no. 5, pp. 43-47, 2013.

[10] Ester. Martin, H. P. Kriegel dan J. Sander, "A Density-Based Algorithm for Discovering *Cluster*s in Large Spatial Databases with Noise," *International Conference on Knowledge Discovery and Data Mining (KDD-96),* vol. II.