5
Cenários Analisados

O objetivo deste trabalho é analisar cenários de utilização de redes neurais artificiais do tipo MLP como detectores de padrões de ataques e intrusões em redes TCP/IP. Com base nos resultados obtidos, espera-se empregar as redes que apresentarem melhores resultados em um comitê de redes especialistas que apresente resultado ainda superior ao resultado de qualquer uma das redes individuais. Finalmente, espera-se verificar que a capacidade de generalização das redes neurais permitirá ao sistema de detecção detectar padrões de ataques e intrusões nunca antes vistos.

Uma grande dificuldade em se estudar cenários de utilização de redes neurais é a necessidade de se possuir uma base de dados para o treinamento supervisionado das redes. Em 1998 a agência norte-americana DARPA financiou um projeto desenvolvido pelo laboratório Lincoln do Massachussets Institute of Technology cujo objetivo era a criação de uma base de dados de referência para avaliação e testes de sistemas de detecção de intrusos. Este trabalho foi extendido e complementado em 1999. A base de dados gerada foi utilizada na competição internacional de mineração de dados de 1999. Os cenários analisados e estudados neste trabalho usaram esta base de dados de referência.

5.1.
Projeto MIT/DARPA 1998-1999

Uma rede de computadores, representativa de uma rede típica encontrada em agências governamentais norte-americanas foi simulada e todo o tráfego de dados capturado. A figura 5.1 apresenta um esquema simplificado desta rede. A rede estudada consiste em 2 segmentos de rede padrão Ethernet interconectados por um roteador Cisco. Geradores de pacotes por software foram empregados gerando tráfego representativo de mais de 20 diferentes serviços de rede, como DNS, FTP, http, PING, POP, SMTP, Telnet, X, Finger etc. Para assegurar a precisão do modelo gerado, estatísticas referentes ao

tráfego de rede e serviços presentes foram coletadas por vários meses em 50 redes diferentes da Força Aérea norte-americana. Os geradores de pacotes foram então ajustados para gerar tráfego com o mesmo comportamento estatístico destas redes reais. Foi capturado tráfego equivalente ao funcionamento de nove semanas desta rede, incluindo de maneira identificada padrões de ataques e intrusões pertencentes as quatro categorias descritas no capítulo 3.

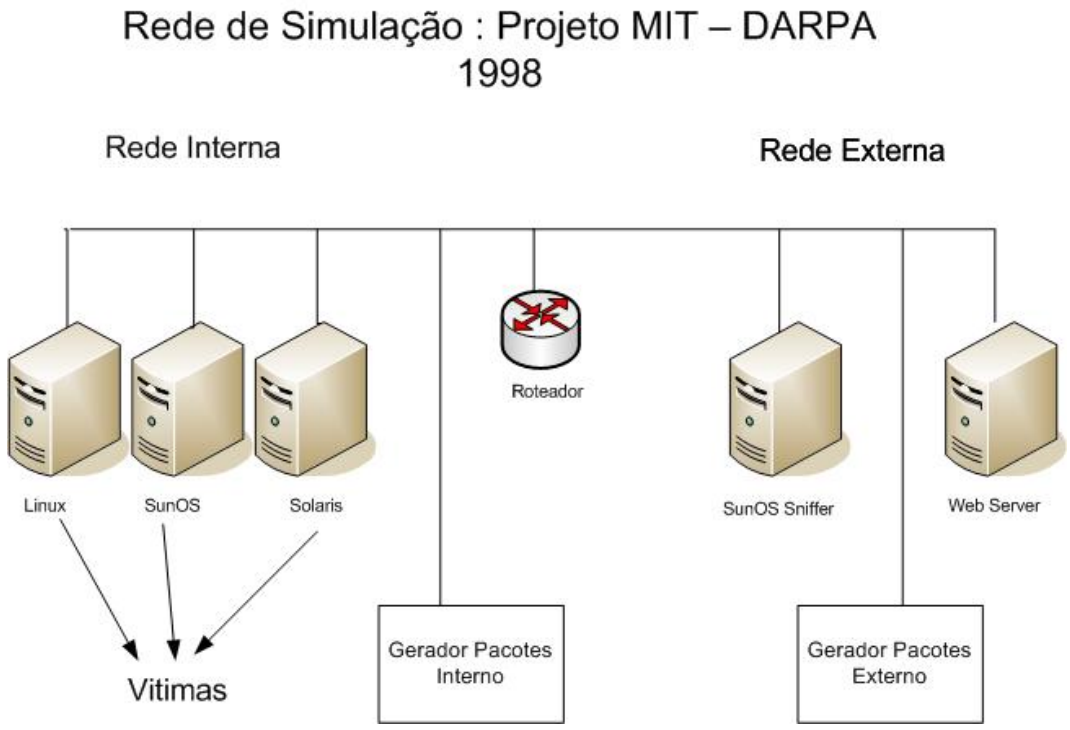

Figura 5-1 – Diagrama da Rede de Simulação

5.2. Base de dados para treinamento e testes

Os dados capturados nas nove semanas consistiam de todos os pacotes IP, TCP, UDP, ICMP capturados nesta rede em formato *tcpdump* [31]. Em 1999 os organizadores da competição internacional em algoritmos de mineração de dados KDDCUP [32] selecionaram esta base de dados e o problema de detecção de intrusão como tema de seu concurso anual. Decidiram, entretanto, realizar uma etapa de pré-processamento na base de dados, reduzindo-a para uma base de dados de registros de conexão. Toda seqüência de pacotes entre um mesmo par de endereços IP e portas de origem e destino específicos, dentro de um intervalo de tempo predefinido tiveram suas principais características

extraídas e condensadas em um único registro de conexão com 41 campos. O programa *tcptrace* [33] realiza esta consolidação de vários pacotes em um arquivo capturado com *tcpdump* [31] em registros de conexão individuais. Cada registro de conexão recebeu ainda um campo identificador classificando-a como uma conexão normal ou um dos

Características clássicas de uma conexão em redes TCP/IP formam o primeiro grupo de campos gerados. Foram denominadas características intrínsecas do TCP/IP e estão apresentadas na tabela 5.2.

Nome	Descrição	Tipo
Duration	Duração em segundos da conexão	Contínua
Protocol_type	Tipo de protocolo usado na conexão, i.e. tcp, udp, etc.	Discreta
Service	Serviço de rede sendo utilizado, i.e. http, telnet, ftp etc.	Discreta
Src_bytes	Número de bytes enviados da fonte para o destino	Contínua
Dst_bytes	Número de bytes enviados do destino para a fonte	Contínua
Flag	Status da conexão (normal ou erro)	Discreta
Land	1 se conexão é de/para o mesmo host; 0 caso contrário.	Discreta
Wrong_fragment	Número de fragmentos com erro.	Contínua
Urgent	Número de pacotes com flag urgente habilitado.	Contínua

Tabela 5-1 : Características intrínsecas de conexões TCP/IP.

Características dependentes de conhecimento especialista humano também foram extraídas da base de pacotes TCP/IP e transformadas em alguns campos nos registros de conexão. Estas características foram obtidas através de análise de informações contidas, primordialmente, na área de dados dos pacotes TCP,UDP e IP. Nesta área, encontra-se normalmente, cabeçalhos de aplicações de nível superior como Telnet, FTP, http, rlogin etc.

Nome	Descrição	Tipo
Hot	Número de indicadores chaves (“hot”).	Contínua
Num_failed_logins	Tentativas de login sem sucesso	Contínua
Logged_in	1 se login efetuado com sucesso; 0 caso contrário.	Discreta
Num_compromised	Número de condições de “comprometimento”	Contínua
Root_shell	1 se shell root foi obtido; 0 caso contrário	Discreta
Su_attempted	1 se comando “su root” foi tentado; 0 caso contrário	Discreta
Num_root	Número de acessos como root.	Contínua
Num_file_creations	Número de operações de criação de arquivos.	Contínua
Num_shells	Número de “shells prompts” obtidos.	Contínua
Num_access_files	Número de operações em arquivos de controle de acesso.	Contínua
Num_outbound_cmds	Número de comandos externos em uma sessão ftp.	Contínua
Is_hot_login	1 se o login pertence a lista “hot”; 0 caso contrário.	Discreta
Is_guest_login	1 se o login usou a conta guest; 0 caso contrário.	Discreta

Tabela 5-2 : Características de conexão por conhecimento especialista

Os campos restantes foram gerados aplicando uma janela de 2 segundos nos pacotes capturados. Esta visão das conexões em relação ao tempo é especialmente importante para a detecção de técnicas de intrusão e ataques das classes de negação de serviço e reconhecimento.

Nome	Descrição	Tipo
Count	Número de conexões iguais a esta para este mesmo “host” nos últimos 2 segundos.	Contínua
Srv_count	Número de conexões para o mesmo serviço que o usado nesta conexão nos últimos 2 segundos.	Contínua
Serror_rate	% de conexões que possuem erro SYN.	Contínua
Srv_serror_rate	% de conexões que possuem erro SYN para este serviço.	Contínua
Rerror_rate	% de conexões que possuem erro REJ.	Contínua
Srv_rerror_rate	% de conexões que possuem erro REJ para este serviço.	Contínua
Same_srv_rate	% de conexões para um mesmo serviço.	Contínua
Diff_srv_rate	% de conexões para serviços diferentes.	Contínua
Srv_diff_host_rate	% de conexões deste mesmo serviço para hosts diferentes.	Contínua
Dst_host_count	Número de conexões com mesmo host de destino que esta.	Contínua
Dst_host_srv_count	Número de conexões com mesmo host de destino e mesmo serviço que esta.	Contínua
Dst_host_same_srv_count	% de conexões com o mesmo host de destino e mesmo serviço que esta.	Contínua
Dst_host_diff_srv_rate	% de conexões com o mesmo host de destino e services diferentes que esta.	Contínua
Dst_host_same_src_port_rate	% de conexões com o mesmo host de destino e a mesma porta de origem que a conexão atual.	Contínua
Dst_host_srv_diff_host_rate	% de conexões para o mesmo	Contínua

	serviço vindo de diferentes hosts.	
Dst_host_serror_rate	% de conexões para o mesmo host que o da conexão atual e que possui um erro S0.	Contínua
Dst_host_srv_serror_rate	% de conexões para o mesmo host e serviço que o da conexão atual e que possui um erro S0.	Contínua
Dst_host_rerror_rate	% de conexões para o mesmo host que apresentem flag RST.	Contínua
Dst_host_srv_rerror_rate	% de conexões para o mesmo host e serviço da conexão atual que apresentem flag RST.	Contínua

Tabela 5-3 : Características temporais : janela de 2 segundos.

Exemplo de três registros de conexão presentes nas bases de dados de treinamento e testes com cada um dos 41 campos separados por vírgula :

0,udp,private,SF,105,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0.00,0.00,0.00,0.00,1.00,0.00,0.00,255,254,1.00,0.01,0.00,0.00,0.00,0.00,0.00,0.00,normal.

0,tcp,http,SF,223,185,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,4,4,0.00,0.00,0.00,0.00,1.00,0.00,0.00,71,255,1.00,0.00,0.01,0.01,0.00,0.00,0.00,0.00,normal.

0,icmp,ecr_i,SF,1032,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,511,511,0.00,0.00,0.00,0.00,1.00,0.00,0.00,255,255,1.00,0.00,1.00,0.00,0.00,0.00,0.00,0.00,smurf.

Os registros de conexões foram divididos em duas bases de dados distintas : a primeira, contendo o equivalente a sete semanas de operação da rede modelo da figura 5.1 (4.898.430 registros), foi denominada base de dados para treinamento dos algoritmos de detecção. A segundo base de dados, contendo o equivalente a duas semanas de funcionamento da rede (311.029 registros) foi denominada base de dados para testes e validação. As tabelas 5.4 e 5.5 apresentam a distribuição dos registros presentes nos dois subconjuntos, incluindo a categoria na qual cada registro está enquadrado. A base de dados de testes e validação possui 15 padrões de ataque que não existem na base de treinamento.

Ataques / Intrusões	# Ocorrências	Percentual	Categoria
spy.	2	0.00004%	R2L
perl.	3	0.00006%	U2R
phf.	4	0.00008%	R2L
multihop.	7	0.00014%	R2L
ftp_write.	8	0.00016%	R2L
loadmodule.	9	0.00018%	U2R
rootkit.	10	0.00020%	U2R
imap.	12	0.00024%	R2L
warezmaster.	20	0.00041%	R2L
land.	21	0.00043%	DOS
buffer_overflow.	30	0.00061%	U2R
guess_passwd.	53	0.00108%	R2L
pod.	264	0.00539%	DOS
teardrop.	979	0.01999%	DOS
warezclient.	1020	0.02082%	R2L
back.	2203	0.04497%	DOS
nmap.	2316	0.04728%	Probing
portsweep.	10413	0.21258%	Probing
ipsweep.	12481	0.25480%	Probing
satan.	15892	0.32443%	Probing
normal.	972780	19.85902%	Normal
neptune.	1072017	21.88491%	DOS
smurf.	2807886	57.32216%	DOS
Total	**4898430**	**100.00000%**	

Tabela 5-4 : Distribuição da base de treinamento

Ataque / Intrusão	# Ocorrências	Percentual
imap.	1	0.00032%
sqlattack.	2	0.00064%
loadmodule.	2	0.00064%
phf.	2	0.00064%
udpstorm.	2	0.00064%
perl.	2	0.00064%
worm.	2	0.00064%
ftp_write.	3	0.00096%
xsnoop.	4	0.00129%
xlock.	9	0.00289%
land.	9	0.00289%
teardrop.	12	0.00386%
xterm.	13	0.00418%
rootkit.	13	0.00418%
ps.	16	0.00514%
named.	17	0.00547%
sendmail.	17	0.00547%
multihop.	18	0.00579%
buffer_overflow.	22	0.00707%
nmap.	84	0.02701%
pod.	87	0.02797%
httptunnel.	158	0.05080%
ipsweep.	306	0.09838%
portsweep.	354	0.11382%
saint.	736	0.23663%
processtable.	759	0.24403%

apache2.	794	0.25528%
mscan.	1053	0.33855%
back.	1098	0.35302%
warezmaster.	1602	0.51506%
satan.	1633	0.52503%
snmpguess.	2406	0.77356%
guess_passwd.	4367	1.40405%
mailbomb.	5000	1.60757%
snmpgetattack.	7741	2.48884%
neptune.	58001	18.64810%
normal.	60593	19.48146%
smurf.	164091	52.75746%
Total	**311029**	**100.00000%**

Tabela 5-5 : Distribuição da base de testes

5.3. Pré-processamento da base de dados

Os dados de treinamento e testes fornecidos pelo projeto MIT/DARPA e KDDCup 1999 não estavam preparados para serem utilizados no treinamento de uma rede neural. Uma etapa de pré-processamento adicional foi realizada. Esta etapa consistiu na transformação de todos os registros em valores numéricos, normalizados no intervalo [-1,1]. A figura 5.2 demonstra uma visão macro de todas as etapas realizadas no processo de captura e preparação dos dados de treinamento e teste das redes neurais simuladas. A monitoração e captura dos pacotes utilizando o software tcpdump [29] foi realizada pelo projeto MIT/Darpa. A primeira etapa de pré-processamento, juntamente com a etapa de extração dos conjuntos de características (temporais, intrínsecas e especialistas) foi realizada pelos organizadores do KDDCup 1999.

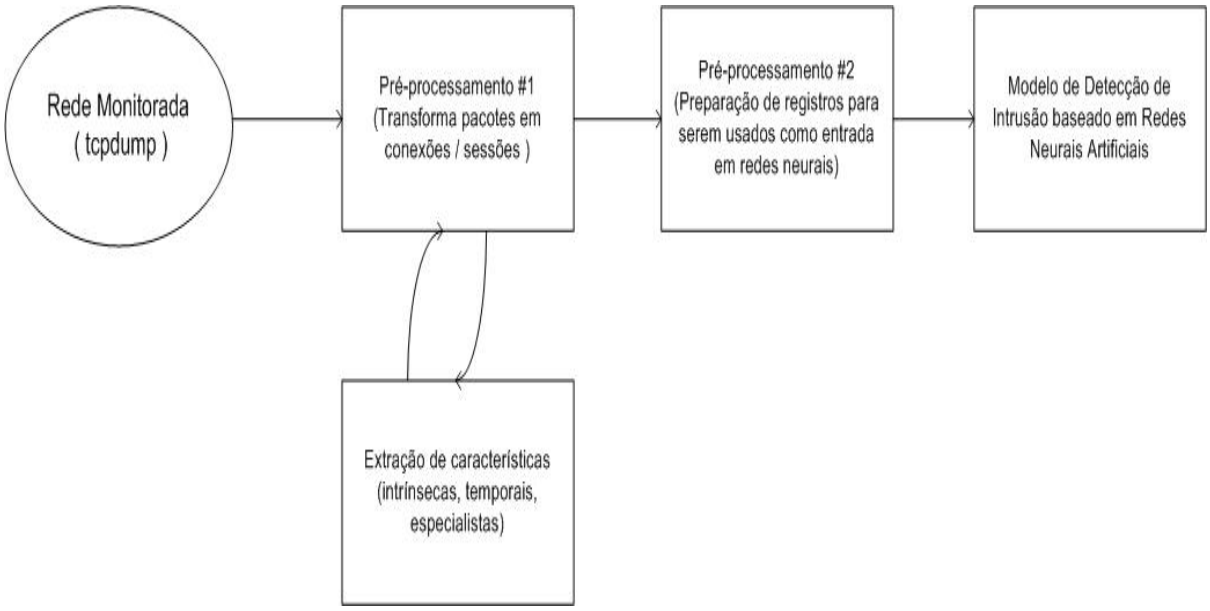

Figura 5-2 : Pré-processamento dos dados

A segunda etapa de pré-processamento foi parte integrante do escopo deste trabalho e consistiu das seguintes etapas :

- Criação de banco de dados relacional – foi empregado o software Microsoft SQL Server [34] - com uma única tabela (denominada kddcup) para armazenar todos os registros de conexão compostos dos 41 campos apresentados nas tabelas 5.1, 5.2 e 5.3.
- Desenvolvimento de scripts em linguagem SQL – ver apêndice E - para transformação de todos os campos não numéricos em campos numéricos.
    - Tabelas temporárias (5.6, 5.7, 5.8 e 5.9) – *flags*, protocolos, *labels* e serviços,– associadas aos campos de mesmo nome presentes na base de dados foram criadas. Cada uma destas tabelas tinha como função associar identificadores numéricos únicos para todo valor não numérico presente nos dados.
    - Substituição dos campos não numéricos presentes na tabela kddcup pelo valor numérico corresponde através de consulta as tabelas temporárias.

Flag	Id.
REJ	1
RSTR	2
S3	3
S1	4
OTH	5
SF	6
S2	7
S0	8
RSTOS0	9
SH	10
RSTO	11

Tabela 5-6 : Flags

Id.	Protocolo
1	icmp
2	tcp
3	udp

Tabela 5-7 : Protocolos

Id.	Label	Categoria de Ataque
1	warezmaster.	r2l
2	guess_passwd.	r2l
3	multihop.	r2l
4	satan.	probe
5	back.	dos
6	buffer_overflow.	u2r
7	loadmodule.	u2r
8	teardrop.	dos
9	perl.	u2r
10	rootkit.	u2r
11	imap.	r2l
12	spy.	r2l
13	land.	dos
14	ftp_write.	r2l
15	pod.	dos
16	smurf.	dos
17	warezclient.	r2l
18	neptune.	dos
19	nmap.	probe
20	ipsweep.	probe
21	phf.	r2l
22	portsweep.	probe
23	normal.	normal

Tabela 5-8 : Labels

Id.	Serviço	Id.	Serviço	Id.	Serviço
1	imap4	25	mtp	49	ftp_data
2	auth	26	csnet_ns	50	hostnames
3	http	27	discard	51	printer
4	efs	28	urh_i	52	urp_i
5	http_2784	29	eco_i	53	private
6	red_i	30	netbios_ssn	54	gopher
7	nnsp	31	uucp	55	time
8	rje	32	courier	56	ctf
9	sql_net	33	pm_dump	57	iso_tsap
10	supdup	34	telnet	58	smtp
11	bgp	35	link	59	IRC
12	daytime	36	tim_i	60	harvest
13	netstat	37	pop_3	61	tftp_u
14	http_443	38	finger	62	netbios_dgm
15	aol	39	pop_2	63	ssh
16	sunrpc	40	ntp_u	64	exec
17	systat	41	vmnet	65	whois
18	ldap	42	netbios_ns	66	ftp
19	kshell	43	domain_u	67	ecr_i
20	shell	44	uucp_path	68	remote_job
21	Z39_50	45	nntp	69	klogin
22	X11	46	http_8001	70	login
23	other	47	domain	71	icmp
24	echo	48	name		

Tabela 5-9 : Serviços

- Geração de cinco subconjuntos – denominados *ds1, ds2, ds3, ds4 e ds5* e armazenados em tabelas de mesmo nome - de dados a partir da tabela original kddcup.
- Os subconjuntos foram gerados através de seleção aleatória de aproximadamente 10.000 registros cada e respeitando os seguintes critérios de seleção :
    - Subconjunto 1 ou *ds1* : contém padrões de todas as classes de ataque e padrões normais.

- Subconjunto 2 ou *ds2 :* contém padrões normais e da classe "reconhecimento" ou "*probe*"
- Subconjunto 3 ou *ds3* : contém padrões normais e da classe negação de serviço.
- Subconjunto 4 ou *ds4* : contém padrões normais e da classe usuário para super-usuário (U2R).
- Subconjunto 5 ou *ds5* : contém padrões normais e da classe remoto para local (R2L).

Subconjunto	Categorias representadas	Total
DS1	Normal, DoS, U2R,R2L, Reconhecimento	10147
DS2	Normal, Reconhecimento	10000
DS3	Normal, DoS	10019
DS4	Normal, U2R	10052
DS5	Normal, R2L	10999

Tabela 5-10 : Subconjuntos de Treinamento.

Para todos os cinco subconjuntos de treinamento, destacados na tabela 5.10, 20% dos registros de conexões foram reservados para realização de procedimentos de validação cruzada conforme apresentado na seção 4.4.1.

5.4. Treinamento e Simulação das redes neurais

Após as etapas de pré-processamento dos dados de teste e verificação vários cenários foram simulados através do módulo de redes neurais artificiais do software Matlab. A tabela 5.11 apresenta os diversos cenários simulados, destacando quais campos foram utilizados como entrada da rede neural e qual a resposta esperada da mesma.

Entrada / **Saída**	Características Intrínsecas do TCP/IP	Características especialistas	Características temporais (janela 2 seg.)	Todas as características.
Ataque / Normal	X	X	X	X
Categoria de ataque / Normal				X
Identificação precisa para classe DoS / Normal				X
Identificação precisa para U2R / Normal		X		X
Identificação precisa para R2L / Normal		X		X
Identificação precisa para “probing” / Normal				X

Tabela 5-11 : Cenários analisados para redes MLP

Os procedimentos macro – código fonte está listado no apêndice E - empregados para treinar, testar e medir o desempenho das redes simuladas consistem em :

1. Carregar subconjunto de treinamento – já processado e contendo apenas campos numéricos normalizados no intervalo [-1;1] conforme descrito na seção anterior -como matriz do Matlab.
2. Definir a rede neural como MLP, determinando o tamanho de vetor de entradas e de saída e estimar o número de camadas e neurônios adequados para o problema. Foi empregada a heurística $\mathbf{N_{hidden} = 2N_{in} + 1}$ com ajustes baseado em tentativa e erro.
   a. Para uma rede, por exemplo, que receba como entrada apenas os 9 campos que compõem as características intrínsecas do TCP/IP a heurística determina que N_{hidden} = 2x9 + 1 = 19 neurônios nas camadas escondidas. Neste caso foi usado na prática uma rede com duas camadas escondidas, cada uma com 10 neurônios.

b. Para a rede que tenha como entrada todas as 41 entradas a heurística recomendaria 83 neurônios distribuídos em camadas escondidas. Experiências com rede com este número de neurônios demonstraram que a rede era capaz de aprender em excesso os vetores de entrada, prejudicando a sua capacidade de generalização. Ajustes, baseados em tentativa e erro, foram realizados para determinar qual o menor número de neurônios atenderiam este problema. O número encontrado empiricamente variou entre 30 e 40 neurônios artificiais distribuídos em duas camadas.

3. Treinar a rede neural com o subconjunto carregado até que uma das seguintes condições seja alcançada :
    a. O erro calculado se torna menor ou igual ao erro desejado.
    b. O erro se estabiliza em determinada faixa de valor, não apresentando queda significativa.
    c. Número máximo de épocas – apresentação de todos os padrões de treinamento para a rede – é alcançado.
    d. Tempo máximo configurado em parâmetro é alcançado.
    e. Erro calculado com 20% dos dados reservados para validação cruzada para de reduzir e apresenta tendência de aumentar seu valor : condição de parada por validação cruzada (figura 4-5).
4. Após o treinamento da rede, os dados de testes e validação são carregados como nova matriz do Matlab.
5. Rede neural treinada é simulada com dados de teste e validação. As saídas da rede são comparadas com saídas reais e os indicadores de desempenho – matriz de confusão e curvas ROC – são determinados.