São Paulo, Brazil
October 29-30, 2018

The Tenth International Conference on
FORENSIC COMPUTER SCIENCE and CYBER LAW

www.ICoFCS.org

DOI: 10.5769/C2018002 or http://dx.doi.org/10.5769/C2018002

Utilização de Redes Neurais Nebulosas para criação de um Sistema Especialista em Invasões Cibernéticas

Lucas Oliveira Batista[1], Gabriel Adriano de Silva[1], Vanessa Souza Araújo[1], Vinícius Jonathan Silva Araújo[1], Thiago Silva Rezende[1] , Augusto Junio Guimarães[1], Paulo Vitor de Campos Souza[1,2]

(1) Faculdade Una de Betim, Email: lobatista@outlook.com.br, adriano.gabriel7@gmail.com, v.souzaaraujo@yahoo.com.br, vinicius.j.s.a22@hotmail.com, silvarezendethiago@hotmail.com, augustojunioguimaraes@gmail.com
(2) Centro Federal de Educação Tecnológica de Minas Gerais, Email: goldenpaul@informatica.esp.ufmg.br

Abstract: —O mundo contemporâneo é caracterizado pela sua constante evolução tecnológica, e a cada dia os processos, antes manuais, se tornam informatizados. Dados são armazenados no espaço cibernético, e em consequência deve-se aumentar a preocupação com a segurança desse ambiente. Os ataques cibernéticos são representados por uma crescente escala mundial e se caracterizam como um dos grandes desafios do século. Este artigo tem como objetivos propor um sistema computacional baseado em modelos híbridos inteligentes, que através de regras *fuzzy* possibilita a construção de sistemas especialistas em ataques a dados cibernéticos, com foco no ataque por SQL *Injection*. Os testes foram realizados com bases reais de ataques *SQL Injection* em computadores governamentais, utilizando redes neurais nebulosas. De acordo com os resultados obtidos, a viabilidade de construção de um sistema com base em regras *fuzzy*, com precisão de classificação de invasões cibernéticas dentro da margem do desvio padrão (comparado com o modelo estado da arte na resolução desse tipo de problema) é real. O modelo auxilia os países a se prepararem para proteger suas redes de dados e seus sistemas de informação, além de criar oportunidades de sistemas especialistas automatizarem a identificação de ataques no espaço cibernéticos.

Key words: Espaço cibernético, Defesa cibernética, Segurança da informação, Redes neurais nebulosas, SQL *Injection*.

I. Introdução

A corrida tecnológica é um dos tópicos mais abordados em todo o mundo atualmente, gerando grandes ferramentas para o comércio, industrias, automação, comunicação, escolas, serviço militar, governos, entre outros setores importantes da economia [1]. Com isso, a busca por alta velocidade de processamento e transmissão das informações e alto desempenho de sistemas, tende a caminhar paralelamente com a segurança e integridade dos dados, fazendo com que o trafego das informações se torne perigoso e alvo de constantes ataques por *hackers*. Em paralelo a este crescimento tecnológico, identifica-se

também uma grande uma evolução dos chamados ataques cibernéticos, procedimentos que visam comprometer a segurança da informação e de sistemas computacionais [2].

Com a globalização e o aumento da dependência da sociedade em relação a sistemas de software, as informações e dados de suma importância para empresas e pessoas trafegam mundialmente de forma instantânea via internet, chamando atenção de criminosos cibernéticos, onde estes buscam invadir sistemas ou interceptar as informações para uso em benefício próprio, ou prejudicial às organizações a quem atacam, tornando os impactos de ataques como esses cada vez mais elevados [3].

Esse artigo tem como proposta a utilização de modelos híbridos baseados em redes neurais e sistemas nebulosos para construir sistemas especialistas em invasão cibernética, baseados em regras *fuzzy*. A invasão que será alvo dessa pesquisa será a detecção realizada por meio do SQL Injection [20], permitindo que o modelo de redes neurais nebulosas atue de forma similar ao estudo proposto por Demertzis et al [3]. O sistema proposto no artigo será capaz de gerar base de regras dos resultados obtidos através de testes, com a utilização de neurônios lógicos nebulosos. Para evitar o *overfitting* e auxiliando na definição da topologia da rede, serão utilizados modelos de treinamento baseado em máquina de aprendizado extremo e na teoria da regularização buscando encontrar os neurônios mais significativos ao problema de invasão cibernética.

A utilização de redes neurais nebulosas vem sendo empregada em diversos ramos, como economia [4], para o reconhecimento de faces humanas na forma 3D [5], seleção de características [6], previsão de vazão de chuvas [7]. Essa técnica inteligente já é utilizada no ramo da segurança da informação como metodologia para detecção de ataques no tráfego de redes [8], e no reconhecimento de padrões de ataque [9]. O emprego da rede neural nebulosa proposta em [10] busca realizar testes de classificações binárias, para criar as regras de reconhecimento de sistemas especialistas em ataques cibernéticos por *SQL Injection*.

O artigo se encontra organizado na seguinte forma: Na seção II temos o referencial teórico, com uma definição de conceitos importantes envolvida no desenvolvimento do trabalho. Na seção III são apresentadas a descrição do processo de utilização do modelo híbrido de inteligência artificial para identificação e sistema especialistas em ataques cibernéticos, com detalhamentos de processos e conceitos específicos utilizados. Finalmente na seção IV são apresentadas as conclusões finais.

2. Referencial Teórico

A. *Espaço Cibernético*

O termo espaço cibernético foi empregado pela primeira vez em um romance escrito por Willian Gibson “*Neuromancer*” [13]; podemos considera-lo como a metáfora que descreve o território não físico criado por meios computacionais, notadamente a internet, onde pessoas físicas e jurídicas, isoladamente ou em grupo, integrantes de empresas, órgãos públicos ou governos, podem se comunicar, realizar pesquisas e trafegar dados de maneira geral, valendo-se de Tecnologias da Informação e Comunicação (TIC) como suporte para seu funcionamento [11]. As ações no espaço cibernético classificam-se em ofensivas, exploratórias ou de proteção, sendo que as ofensivas podem impactar, até mesmo, a segurança nacional [12].

B. *Ataques Cibernéticos*

Na situação mundial o espaço cibernético é uma área na qual, apesar de se ter a compreensão da necessidade de segurança, não existem medidas implementadas de maneira sistemática e articulada que possam garantir a confiabilidade e a preservação dos sistemas empregados. De maneira incipiente, as nações vêm se preparando para evitar ou minimizar ataques cibernéticos às redes e sistemas de informação de governo, bem como de todos os demais segmentos da sociedade [11].

Os ataques podem acontecer de maneira física, onde os dispositivos contendo as informações são de fácil acesso, além de modens, cabos e mídias de armazenamento físicas [16]. Pelo meio humano, o ataque é empregado pela engenharia social, e pelo meio lógico, em que são empregadas técnicas como a de invasão para derrubar serviços (DDoS) onde o atacante trabalha para sobrecarregar o sistema em foco.

Técnicas que exploram vulnerabilidade de portas de acesso, a disparada de vírus e malwares, ou até decodificadores de senhas, este último que consiste em um script que tenta decifrar senhas [16].

Outro aspecto importante a ser considerado se relaciona aos crimes cibernéticos, particularmente devido aos efeitos danosos que podem advir do mau uso dos sistemas de informação e comunicação por pessoas mal-intencionadas. Apesar dos esforços de alguns setores da Administração Pública, ainda há brechas na legislação brasileira, não havendo leis para alguns tipos de ações que já são consideradas crime em outros países. Além disso, não há uma política clara embasando uma ação contra outro país que tenha afetado de alguma forma infraestrutura crítica nacional por meios cibernéticos e muitas questões ainda estão merecendo algum tipo de tratamento, sendo que esse avanço está sendo mais lento do que a situação exige [12].

C. *SQL Injection*

Structured Query Language, ou simplesmente SQL, é a linguagem padrão para fazer interação com banco de dados relacionais. Nela podemos fazer as principais tarefas relacionadas a manipulação de dados em estruturas de banco de dados [20].

SQL Injection é um tipo de ataque cibernético que se aproveita de falhas em sistemas que normalmente tem uma comunicação com base de dados através de comandos SQL, e por esse motivo é considerado um tipo de ataque é muito simples. Nesse processo de invasão o atacante consegue inserir uma instrução SQL personalizada e indevida dentro de uma consulta (*SQL query*) através das entradas de dados de uma programa, como formulários ou URL de uma aplicação. Nos campos destinados a informação do usuário, esses comandos são realizados, ou seja, são exibidos comandos SQL, todavia por motivo dessa falha nas aplicações acabam ocasionando alterações no banco de dados ou em acesso inadequado a aplicação [21].

Um *cracker* consegue obter qualquer tipo de dado sigiloso mantido no banco de dados de um computador servidor, por meio de ataques com injeção SQL, inclusive dependendo da versão do banco de dados, também é possível inserir comandos maliciosos e conseguir permissão total à máquina em que o banco está em execução [22]. A figura 1 exemplifica o ataque de *SQL Injection* em um formulário.

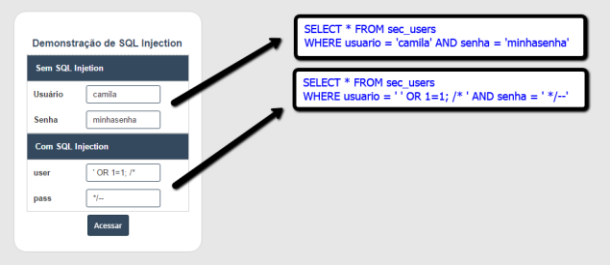

Fig. 1. Exemplo de *SQL Injection* realizado em um formulário de dados. Disponível em: *https://www.scriptcaseblog.com.br/sql−injection− injetando − dados − a − partir − de − inputs/*

D. *Defesa Cibernética*

De acordo com as definições de Mandarino: “Segurança cibernética e entendida como a arte de assegurar a existência e a continuidade da Sociedade da Informação de uma nação, garantindo e protegendo, no Espaço Cibernético, seus ativos de informação e suas infraestruturas críticas” [14]. Também podemos caracterizá-la como um conjunto de ações defensivas, exploratórias e ofensivas, no contexto de um planejamento, realizadas no espaço cibernético, com as finalidades de proteger os nossos sistemas de informação [12].

Manter os dados virtuais seguros está sendo um desafio século XXI para o Brasil, tem-se um destaque cada vez maior na segurança cibernética, como função estratégica de estado, sendo essencial à manutenção do funcionamento das infraestruturas críticas do país. Pode-se afirmar que o país para se desenvolver não pode abdicar da segurança do seu espaço cibernético. De acordo com o que está previsto na Estratégia Nacional de Defesa (END) [17], a responsabilidade de coordenar as ações de Defesa Cibernética fica a encargo do Exército, no âmbito das Forças Armadas.

Devido a essa responsabilidade, o Exército Brasileiro está criando o Centro de Defesa Cibernética, vem se articulando nesta área e consolidando sua posição [12]. Observa-se, portanto, que existe a noção da importância de se tomar ações que favoreçam e permitam

estabelecer a segurança no espaço cibernético, embora não se tenha a exata dimensão do que isso possa representar, bem como quais as medidas que efetivamente podem e devem ser adotadas para se atingir este objetivo.

E. *Redes neurais artificiais*

As redes neurais artificiais são modelos inteligentes que utilizam em suas estruturas o neurônio lógico, buscando simular o processamento de informação do cérebro humano através de uma rede de diversos neurônios artificiais interligados, que se unem por meio de conexões sinápticas. De uma forma simplificada, uma rede neural artificial pode ser vista como um grafo onde os nós são os neurônios e as ligações fazem a função das sinapses [18].
As redes neurais artificiais se diferenciam pela sua arquitetura e pela forma como os pesos associados às conexões são ajustados durante o processo de aprendizado. O aprendizado é a forma com que a rede neural capta as informações fornecidas pelas entradas e através das conexões e dos pesos sinápticos tomam decisões acerca do tema central da base de dados. A arquitetura de uma rede neural restringe o tipo de problema no qual a rede poderá ser utilizada, e é definida pelo número de camadas (camada única ou múltiplas camadas), pelo número de nós em cada camada, pelo tipo de conexão entre os nós (*feedforward* ou *feedback*) e por sua forma de atuação [18]. A figura 2 apresenta a estrutura de uma rede neural com múltiplas camadas. Ela também evidencia os neurônios e suas ligações sinápticas.

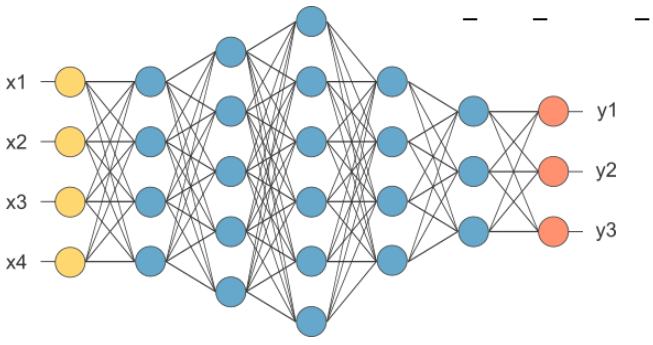

Fig. 2.Exemplo de uma rede neural artificial de múltiplas camadas e múltiplas saídas. Disponível em:*http://www.decom.ufop.br/imobilis/fundamentosderedes neurais/*

F. *Sistemas Nebulosos*

O uso dos sistemas nebuloso faz-se necessários em casos onde a abordagem lógica clássica torna-se inviável para a resolução de um problema devido à natureza de sua complexidade [19]. Os métodos mais conhecidos são passíveis de alterações bruscas para a resolução de problemas devido à simplificação do modelo real, porém os sistemas nebulosos possuem recursos (funções de pertinência, regras e operadores de agregação) que possibilitam a aproximação mais fiel ao modelo real, evitando que a solução gerada pelo sistema nebuloso destoe consideravelmente do modelo real. A figura 3 apresenta os principais elementos que compõem a lógica *fuzzy*: suas entradas, o processo da transformação de entradas em elementos *fuzzy*, a criação de conjuntos *fuzzy* de entrada, o conjunto de regras e inferências, a obtenção dos conjuntos *fuzzy* de resposta, a defuzzyficação que é tornar os valores obtidos de acordo com as entradas do sistema e as saídas de forma esperada.

Fig. 3. Conceitos presentes na lógica nebulosa (fuzzy). Disponível em: *https://www.devmedia.com.br/introducao-a-logica-fuzzy-com-java/*32444

G. *Redes neurais fuzzy - Conceitos Gerais*

Redes neurais *fuzzy* usam a estrutura de uma rede neural artificial (RNA), onde clássicos de neurônios artificiais são substituídos pelos neurônios nebulosos [7] [9]. Estes neurônios são implementados por meio de normas triangulares que generalizam as operações de União e interseção de conjuntos clássicos permitindo que elas sejam utilizadas na teoria dos conjuntos *fuzzy*. Assim, a rede neural é agora vista como um sistema linguístico, preservando a capacidade de aprendizado da RNA [8].

Eles fornecem uma rede como a topologia e permite o uso de uma ampla variedade de processos de aprendizagem com bases de dados de diversos contextos. A principal característica dessas redes é a sua transparência, permitindo o uso de informações a priori para definir a topologia de rede inicial e permitindo a extração de informações valiosas da topologia resultante após o treinamento sob a forma de um conjunto de regras *fuzzy* [10].

A atualização de parâmetros, tipos de neurônios utilizados e o algoritmo de treinamento divergem em relação aos topologias de rede criadas. Existem redes neurais fuzzy que utilizam máquina de aprendizado extremo [26] para atualizar parâmetros internos das RNN onde os seus neurônios lógicos são unineurons [24], nullneurons [25], e andneurons [10].

2. Metodologia de Identificação de Ataques *SQL INJECTION* (bioHAIFCS)

O artigo sobre detecção de anomalias de rede com base em evolução de rede neural escrito por Konstantínos Demertzís e Lazaros Iliadis [3], descreve um sistema inteligente de aprendizagem de máquina, onde parte do sistema trabalha procurando ameaças conhecidas, e outra parte tenta detectar prováveis ameaças de acordo com atividades anormais que acontecem no sistema. O sistema de detecção é basicamente simples, ele gera um estado sendo tratado como normal, e todos os sinais fora da margem desse estado são tratados como anomalia, assim o algoritmo de detecção aprende continuamente enquanto o sistema está ativo em rede, sendo cada vez mais preciso.

A metodologia utilizada no artigo foi *a Spiking Artificial Neural Networks* (SANN) [3], que utiliza uma abordagem de classificação *Evolving Connectionist System* (eCOS) e *Multi-Layer Feed Forward* ANN para classificar o tipo exato da invasão ou anormalidade na rede com mínimo potencial computacional. SANN é um conjunto de sistemas modulares baseados em conexões em nó. O sistema se auto organiza de maneira continua, em modo linha, se adaptando a partir dos dados de entrada, podendo funcionar ou não de maneira supervisionada. O SANN também está sendo aplicado a vários outros problemas complexos do mundo real, se mostrando bastante eficiente.

O nome do modelo desenvolvido se chama bio HAIFCS(*Hybrid Evolving Spiking Anomaly Detection Model*), este que trabalha em cima dos picos que ocorrem no sistema, enquanto a os neurônios são usados para monitorar o algoritmo utilizando aprendizagem *OnePass*. São usados dados orientados ao trafego, importando as classes, estas que usam a variável *Population Encoding* (variável de controle de conversão de dados da amostra para o valor real nos picos de tempo). Os dados foram classificados em dois tipos, Classe 0, é a classe dos resultados normais. Classe 1, correspondente aos resultados anormais. Quando há verificação e o resultado é 0, o processo de classificação eSNN é repetido, mas com vetores de dados relevantes. Caso o resultado continue 0, o processo é finalizado. Quando o resultado é Classe 1, usa-se uma rede neural de duas camadas para reconhecer o padrão do tipo de ataque, utilizando todos os recursos da base de dados do KDD, se acontecer na camada oculta, são utilizados 33 neurônios. Os resultados do processo são apresentados ao administrador da rede em forma de alerta, o modelo gráfico do HESADM pode ser analisado na figura 4 [3].

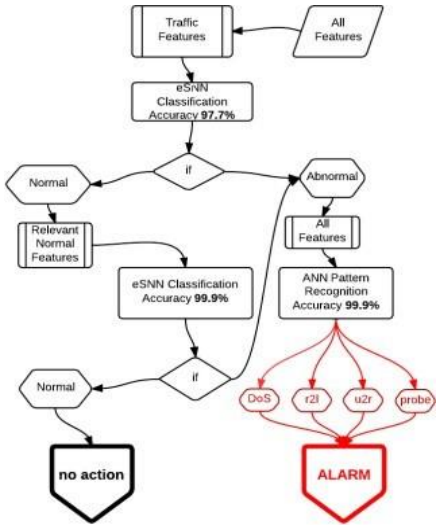

Fig. 4. Modelo HESADM [3]

O conjunto de dados utilizados para testes foram do Copa KDD 1999, criados no *LincolnLab* do MIT [23], conjunto este que segundo os autores do

HESADM é o mais popular utilizado. O conjunto contém dados que simulam uma rede das forças aéreas dos EUA. “O método de análise de eventos inclui uma conexão entre um endereço IP de origem e um IP de destino, durante o qual uma sequência de pacotes TCP é trocada, usando um protocolo específico e um tempo de operação estritamente definido. O banco de dados usado inclui uma lista de 13884 instruções SQL que foram selecionadas por várias fontes. Na verdade, 12881 deles são maliciosos (*SQL Injections*) e 1003 são legítimos. a correlação de SQLstatements com o tipo de injeções de SQL. Finalmente, a técnica *n-gram* foi usada para pesquisar a correlação da seqüência de instruções SQL, com a sintaxe dos comandos de injeção de SQL [3].

Na figura 5 o bom desempenho e a confiabilidade do esquema proposto em [3] são mostrados. Ela apresenta os resultados da categorização com o mesmo conjunto de dados de SQL Injection e empregando Validação Cruzada com 10 k-fold e outras abordagens de Aprendizado de Máquina [3] O modelo chegou ao resultado de 99,6%.

Table 7. *Comparison of various approaches for the SQLI dataset*

SQLI Dataset	
Classifier	**Accuracy**
MFF ANN with GA	99.6%
RBFNetwork	97.3%
NaiveBayes	95.6%
BayesNet	98.7%
SVM	98.5%
k-NN	98,3%
Random Forest	99.1%

Fig. 5. Resultados obtidos pelo modelo híbrido proposto em [3]

4. Rede Neural Fuzzy Utilizada Para Detecção de Ataques SQL-Injection

O modelo para detecção de ataques cibernéticos foi proposto inicialmente para classificação de padrões binários. O modelo apresentado é um agrupamento de conceitos dos modelos propostos em [10], [24] e [25]. A figura 6 apresenta a arquitetura da rede neural nebulosa utilizada nos testes.

As duas primeiras camadas do modelo são consideradas um sistema nebuloso de inferência, capaz de extrair conhecimento dos dados e transformá-los em regras nebulosas. Essas regras auxiliam a construção de sistemas automatizados para detecção de perfil de compra das pessoas de acordo com as respostas fornecidas no teste. Diferentemente abordado em [24] terceira camada é composta por um neurônio simples que tem como função de ativação a metodologia proposta por [27], chamada leaky ReLU.

A primeira camada é composta de neurônios nebulosos cujas funções de ativação são funções de pertinência Gaussianas dos conjuntos nebulosos definidos conforme a partição das variáveis de entrada. Para cada variável de entrada x_{ij} são definidos M conjuntos nebulosos A^m, *m de 1*,..., M cujas funções de pertinência são as funções de ativação dos neurônios correspondentes. Portanto as saídas da primeira

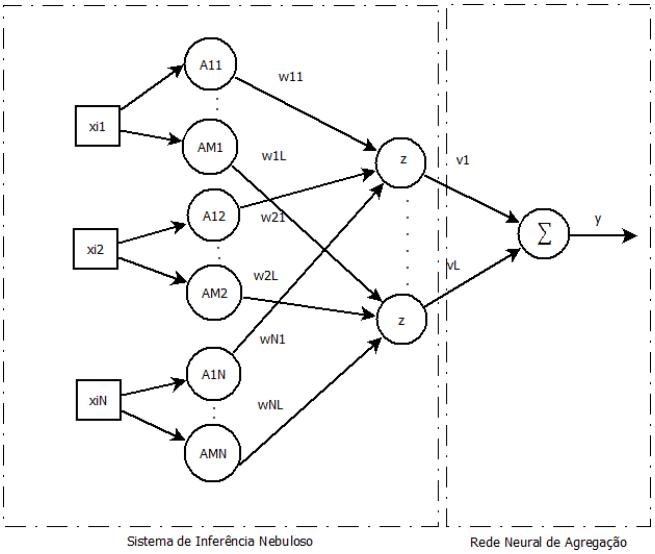

Fig. 6. Arquitetura da rede neural nebulosa [24]

camada são os graus de pertinência associados com os valores de entrada, isto é, $\alpha^j = \mu_{A^m}$ para j=1,..,N e m=1,...,M, onde N *é o número de entradas e* M é o número de conjuntos nebulosos para cada variável de entrada [10, 24, 25]. Para os neurônios da primeira camada os valores de bias e de pesos sinápticos são definidos de forma

aleatória no intervalo de [0, 1]. Nesse trabalho utilizaremos a totalidade de combinações de conjuntos nebulosos gerados para cada variável quando *N* for menor ou igual a 6 [24]. Quando *N* tem altos valores numéricos realizamos a escolha aleatória de uma função de pertinência para cada variável de entrada, onde *M* nesse caso será o dobro do valor de amostras do espaço de entrada, limitadas a 500 funções de pertinência. Em seguida utilizamos as saídas dos neurônios nebulosos do modelo para definir uma quantidade de neurônios candidatos (L_c) que representam um percentual de *L* onde $L_c < L$. Por definição, quando a $L<200$ utiliza-se $L_c = 100$ % de *L*, caso contrário o percentual escolhido é capaz de selecionar os neurônios candidatos. Esse percentual permite a escolha dos neurônios mais significativos da primeira camada [25].

A segunda camada é composta de L_c neurônios lógicos nebulosos, onde destacamos o *unineuron* proposto por [27]. Cada neurônio executa uma agregação ponderada de algumas saídas (e não de todas devido a técnica de seleção de neurônios) da primeira camada juntamente com o bias e os pesos definidos de forma aleatória dos *unineurons* [24]. Considere como sinal de entrada **a** $[a_1; a_2; ...a_n]$ e os pesos **w** $[w_1;w_2;...w_n]$ para $a_i \in [0, 1]$ e $w_i \in [0, 1]$ para *i* de 1,...,*n*. A agregação realizada pelos neurônios lógicos nebulosos *and* e *or* onde os sinais de entrada são combinados individualmente com os pesos e realizada a agregação global posterior podem ser definidas como se segue (Pedrycz, 1993):

$$z = OR(a, w) = S_{i-1}^{n}(a_i t w_i) \quad (1)$$

$$z = AND(a, w) = T_{i-1}^{n}(a_i s w_i) \quad (2)$$

onde *T* e *t* são a representação de uma *t-norma* e *S* e *s* uma *s-norma*. Já para o unineuron [27] descreveram os passos para realizar as funções do neurônio:

• Transformar cada par (a_i,w_i) em um único valor $\mathbf{b}_i=h(a_i,w_i)$;

• Calcular a agregação uninorma dos valores transformados **U** $(b_1,b_2,...,b_n)$, onde *n* é o número de entradas. A função *h* é responsável por transformar as entradas e os pesos correspondentes em valores transformados individuais [27]. Uma formulação para a função *h* pode ser visualizada:

$$h(w, a) = w * a + w * g \quad (3)$$

Utilizando a agregação ponderada relatada em (3) podemos escrever o unineuron [27]:

$$z = UNI(w, x, g) = U_{i-1}^{n} h(w_i x_i) \quad (4)$$

Regras *fuzzy* podem ser extraídas através da topologia da redeem conjunto com os neurônios lógicos unineuron. A seguir um exemplo de combinações sobre as regras formadas:

Regra1: Se x_{i1} é A^1_1 com certeza w_{11}...

e x_{i2} is A^1_2 com certeza w_{21} ...

então y_1 é v_1

Regra2: Se x_{i1} é A^2_1 com certeza w_{12}... (5)

e x_{i2} is A^2_2 com certeza w_{22} ...

então y_2 é v_2

Regra3: Se x_{i1} é A^3_1 com certeza w_{13}…

então y_3 é v_3

Regra4: Se x_{i2} é A^2_2 com certeza w_{23}…

então y_4 é v_4

Após a definição dos neurônios candidatos a arquitetura final da rede é definida utilizando a seleção de um subconjunto desses neurônios. Ao realizar esse procedimento estamos realizando um subconjunto ótimo de valores, podendo ser visualizado como um problema de seleção de variáveis, retornandoos neurônios mais significativos (L_s) baseados em uma função de custo [24]. De forma análoga podemos interpretar essa seleção como a escolha do melhor conjunto de regras capaz de representar o espaço de entrada. O algoritmo de

aprendizagem assume que a saída da segunda camada da rede neural nebulosa composta por todos os neurônios mais significativos (L_s) pode ser escrita como

$$f(x_i) = \sum_{l=0}^{ls} v_l z_l(x_i) = z(x_i)v \quad (6)$$

Onde $\mathbf{v}=[v_0,v_1,v_2,\ldots,v_{Ls}]$ é o vetor de pesos da camada de saída e o $\mathbf{z}(\mathbf{x_i})=[z_0,z_1(x_i),z_2(x_i)\ldots,z_{Ls}(x_i)]$ é o vetor argumento (linha) de saída da segunda camada, para $z_0=1$. Nesse contexto, $z(x_i)$ é considerado como o mapeamento não linear do espaço de entrada para um L_s+1 espaços de características nebulosas dimensionais, executado utilizando os neurônios selecionados. Como os pesos que ligam a as duas primeiras camadas são atribuídos de forma aleatória e os únicos parâmetros estimados foram os pesos da camada de saída podemos visualizar a equação (6) como um modelode regressão linear simples permitindo que o problema da escolha dos melhores subconjuntos de neurônios que serão selecionados possa ser idealizado como um modelo de regressão linear comum para problemas de seleção [29]. Um algoritmo muito utilizado para realizar seleção de modelos foi criado por [30] e é conhecido como Algoritmo *Least Angle Regression* (LARS). O LARS é um algoritmo de regressão para dadoscom altas dimensões que não é capaz de estimar somente os coeficientes de regressão, mas também um subconjunto de regressores candidatos a serem incluídos no modelo final. Ao avaliarmos um conjunto de K amostras distintas (x_i,y_i), onde $\mathbf{x_i}=[x_{i1},x_{i2},\ldots,x_{iN}] \in \mathbb{R}$ e $\mathbf{y_i} \in \mathbb{R}$ para $i=1,\ldots,K$, a função de custo desse algoritmo de regressão pode ser definida como:

$$\sum_{i=1}^{K} \| z(x_i)v - y_i \|_2 + \lambda \| v \|_1 \quad (7)$$

Onde λ é um parâmetro de regularização estimado utilizando o método de validação cruzada. O primeiro termo de (7) corresponde soma residual dos quadrados (RSS). Esse termo diminui a medida que o erro de treinamento também cai. O segundo termo é um termo de regularização l_1. Essa expressão é utilizada pois ele melhora a generalização da rede evitando o superajuste [31] e é capaz de gerar modelos esparsos [32]. Reescrevendo a equação (7) entendemos o porque de LARS ser utilizado como algoritmo de seleção de variáveis:

$$min_v \; RSS(v) \quad (8)$$
$$s.t. \|v\|_1 < B$$

Onde B é um limite superior na l_1-norma dos pesos. Um pequeno valor de B corresponde a um grande valor de λ, e vice-versa. Essa equação também é conhecida como lasso [33]. Quando utilizamos a regressão lasso (também chamadade l_1-norma) para a regularização de modelos, verificamos que o método leva a resultados com soluções espaças, gerando vetores resultantes com muitos zeros, que representam dados sem importância para o resultado das variáveis analisadas, possibilitando uma melhor seleção dos modelos [34]. O algoritmo LARS pode ser utilizado para efetuar a seleção do modelo, uma vez, para um dado valor de λ apenas uma fração (ou nenhuma) dos regressores tem pesos correspondentes diferentes de zero. Se λ= 0, o problema torna-se regressão irrestrita, e todos os pesos são diferentes de zero. À medida que aumenta λ, de 0 a um determinado valor λ_{max}, o número de pesos diferentes de zero diminui a partir de N a 0. Para o problema considerado neste trabalho, os regressores z_{ls} são as saídas dos neurônios significativos. Assim, o algoritmo LARS pode ser utilizado para selecionar um subconjunto ótimo dos neurônios significativos que minimizam (8) para um dado valor de λ, obtido através da validação cruzada. Utilizando o conceito de *bootstrap* e realizando a interseção entre suportes, Bach (2008) desenvolveu um modelo estimador de regularização, sem as condições de consistências exigidas pelo método lasso. A esse novo procedimento ele deu o nome de *Bolasso* (*bootstrap-enhanced least absolute shrinkage operator).* Esse *framework* pode ser visto como um esquema de votação aplicada aos suportes de estimativas do método lasso. No entanto o *Bolasso* pode ser visto como um regime de combinações de consensos onde é

mantido o maior subconjunto de variáveis sobre as quais todos os regressores concordam quando o aspecto é a seleção de variáveis [34]. Os regressores a serem incluídos no modelo final são definidas de acordo com a frequência com que cada um deles é selecionado através de diferentes ensaios. Um limiar de consenso é definido, digamos ρ= 50%, e um regressor está incluído, se for selecionada em, pelo menos, 50% dos ensaios. Nesse artigo o *bootstrap lasso* é utilizado para definir a topologia da rede e escolher os neurônios mais significativos. Os conceitos de máquina de aprendizado extremo [26] são aplicados para calcular os pesos da camada de saída e a rede neural de agregação, presente na terceira camada do modelo realiza a classificação de padrões de ataques cibernéticos conforme a equação (9):

$$y = sign(f_{leaky\,\mathrm{Re}\,lu}(\sum_{j=0}^{ls} z_j v_j)) \quad (9)$$

Onde $z_0 = 1$, v_0 é o bias, e z_j e v_j, $j = 1, ..., l_s$ são a saída de cada neurônio nebuloso da segunda camada e o seu peso correspondente, respectivamente. A função leaky ReLU é expressa por [27]:

$$f_{leakyReLU}(z, \alpha) = \max(\alpha z, z) \quad (10)$$

Essa função de ativação sem sendo empregada em problemas de diversas naturezas, principalmente aqueles onde exige uma sensibilidade maior nos resultados obtidos pelas redes neurais nebulosas.

5. Resultados dos Testes Utilizando Sistemas Especialistas

Para realizar os testes, foi utilizado a base de dados *KDD Cup* 1999, que inclui 13869 casos dos quais 12881 são maliciosos e 988 legítimos (0,0723%) de ataques SQL Injection. Conjuntos de dados desequilibrados são um caso especial para problemas de classificação onde a distribuição de classes não é uniforme entre as classes. Normalmente, eles são compostos por duas classes: A maioria (negativa) e a minoria (positiva). Destas características, foram utilizados os parâmetros *comprimento, entropia, nível de malicia, nível de confiança, diferença de nível e Classe*. O sistema especialista é baseado nas regras SE e ENTÃO. Os modelos de redes neurais nebulosas (UNI-RNN [24] é rede neural fuzzy composta por unineurons (4) e AND-RNN [10] é composta por andneurons (2). Foram comparados com outros algoritmos classificadores para a base de dados: SVM (*Support Vector Machine*), MLP (*Multlayer Perceptron*), NB (*Naive Bayes*) e C4.5. As condições de teste foram similares as procedidas no trabalho realizado em [3], onde as configurações e a utilização da ferramenta weka foram as mesmas. Os resultados (Tabela 1) foram providos dos testes realizados em uma máquina desktop com processador Intel Core i5-3470 3.20GHz e 4,00 GB Memoria.

TABLE I
ACURÁCIA DOS MODELOS PARA A BASE SQL-INJECTION.DATA

Modelos	Acurácia	AUC	Sensib.	Tempo Teste
UNI-RNN	98.44 (0.15)	98.00 (0.01)	98.96 (24.23)	586.14 (10.12)
AND-RNN	98,46 (0.21)	98.00 (0.01)	97.94 (0.37)	771.69 (108.48)
SVM	96.79 (2.71)	96.14 (2.76)	96.87 (1.42)	468.97 (48.76)
MLP	92.44 (7.15)	91.87 (6.03)	87.65 (7.87)	714.18 (56.96)
NB	95.14 (2.14)	62.09 (1.34)	79.65 (5.15)	543.12 (33.02)
C4.5	92.18 (3.43)	96.54 (11.76)	84.36 (3.45)	268.31 (7.65)

A. Interpretabilidade do modelo resultante

O sistema apresentou resultados efetivos sobre a utilização de regras *fuzzy* para a construção de sistemas especialistas. Podemos destacar o exemplo obtido quando se utilizada três funções de pertinência permitindo que os parâmetros sejam classificados em "pequeno, médio e elevado".
Um exemplo de regra obtida pode ser a seguinte: "*Se O comprimento é médio **E**/**OU** a entropia é baixa **E/OU** o nível de malícia é elevado E o Nível de Confiança é pequeno E a diferença de nível é média então existe uma invasão de SQL Injection*."

6. Conclusão

Posteriormente a realização dos testes utilizando o modelo de rede neural nebulosa regularizada, concluímos que os resultados dos experimentos comprovam que a criação de um sistema especialista auxilia na prevenção de ataques

cibernéticos, fomentando a construção de aplicativos inteligentes.

Através dos resultados obtidos, pode-se afirmar que os testes realizados foram satisfatórios. Tendo como base de comparação os resultados obtidos por Demertzis [3], o modelo aqui apresentado é estatisticamente equivalente ao modelo HESADM [3], atingindo um nível próximo do estado da arte, sendo 98% nos testes aqui gerados e 99% da comparação, e apesar de ter sido menor que o artigo comparado [3], nosso modelo possui uma interpretabilidade muito maior fazendo com que pessoas especialistas no assunto validem as soluções do sistema. Consequentemente os resultados também ficam mais interpretáveis e de fácil utilização, além disso é um trabalho que tem um grande valor no âmbito científico.

Referências

[1] Monteiro, R. L. CIBERNÉTICA: A INVASÃO DA PRIVACIDADE E DA INTIMIDADE CYBERNETICS: THE INVA-SION OF PRIVACY AND INTIMACY.

[2] RJ, C., RJ, C. Uso de Workflows Científicos para Apoiar aElaboração de Técnicas de Predição de Invasão de Sistemas.

[3] Demertzis, K., & Iliadis, L. (2015). A bio-inspired hybrid artificial intelligence framework for cyber security. In *Computation, Cryptography, and Network Security* (pp. 161-193). Springer, Cham.

[4] Bakirtzis, A. G., Theocharis, J. B., Kiartzis, S. J., Satsios,K. J. (1995). Short term load forecasting using fuzzy neural networks. IEEE Transactions on Power Systems, 10(3), 1518-1524.

[5] Marin, L. D. O. (2003). Investigações sobre redes neurais artificiais para o reconhecimento de faces humanas na forma3D.

[6] Silva, A. M., Caminhas, W. M., Lemos, A. P.,Gomide,F. (2013, September). Evolving neo-fuzzy neural network with adaptive feature selection. In Computational Intelligence and 11th Brazilian Congress on Computational Intelligence(BRICS-CCI CBIC), 2013 BRICS Congress on (pp. 341-349).IEEE.

[7] Ballini, R., Soares, S., Andrade, M. G. (2003). Previsão devazões médias mensais usando redes neurais nebulosas. Sba:ControleAutomação Sociedade Brasileira de Automatica,14(3), 680-693.

[8] de Sá Silva, L. (2007). UMA METODOLOGIA PARADETECÇÃO DE ATAQUES NO TRÁFEGO DE REDESBASEADA EM REDES NEURAIS (Doctoral dissertation,Tese de Doutorado do Curso de Pós-Graduação em Com-putação Aplicada-Instituto Nacional de Pesquisas Espaciais).

[9] Netto, R. S. (2006). Detecção de Intrusão Utilizando Redes Neurais Artificiais no Reconhecimento de Padrões de Ataque.Universidade Federal de Itajubá, Itajubá.

[10] Souza, P. V. C. (2018). Regularized Fuzzy Neural Networks for Pattern Classification Problems. International Journal ofApplied Engineering Research, 13(5), 2985-2991.

[11] Hosang, A. (2011). Política Nacional de Segurança Cibernética:uma necessidade para o Brasil. Escola Superior De Guerra, RioDe Janeiro.

[12] Carvalho, P. S. M. D. (2011). A defesa cibernética e as in-fraestruturas críticas nacionais. Coleção Meira Mattos-Revistadas Ciências Militares.

[13] Canongia, C.,Mandarino Junior, R. (2010). Segurançacibernética: o desafio da nova Sociedade da Informação. Parcerias Estratégicas, 14(29), 21-46.

[14] MANDARINO JR, R. (2009). Um estudo sobre a segurança e adefesa do espaço cibernético brasileiro. Monografia aprovadano Curso de Especialização em Gestão da Segurança da In-formação e Comunicações. Brasília: Universidade de Brasília-UnB/Departamento de Ciência da Computação, p. 29.

[15] Matsuyama, K. G.,LIMA, J. (2017). Crimes cibernéticos:atipicidade dos delitos.

[16] Duarte, L. O. (2003). Análise de vulnerabilidades e ataquesinerentes a redes sem fio 802.11 x. UNESP–IBILCE–São Josédo Rio Preto.

[17] DE OLIVEIRA, E. R. (2009). A estratégia nacional de defesa e a reorganização e transformação das Forças Armadas. Interesse Nacional, Abril/Junho, 71-83.

[18] Haykin, S., Network, N. (2004). A comprehensive foundation.Neural networks, 2(2004), 41.

[19] Calvo, R. (2007). Arquitetura híbrida inteligente para naveg-ação autônoma de robôs (Doctoral dissertation, Universidadede São Paulo).

[20] Baltazar, L., Valadares, N., Andrew, C. Ataques SQL Injectionem Banco de Dados Através de Aplicações Web: Um Estudo.Publicado por.

[21] Espindola, M. (2017). Segurança em comércio eletrônico.Gerência de Projetos de Tecnologia da Informação-UnisulVirtual.

[22] Vissotto Jr, A., Bosco, E., BRUSCHI, G. C., Silva, L. A.(2016). Prevenção de Ataques: XSS Residente e SQL Injectionem Banco de Dados PostgreSQL em ambiente WEB. Cadernode Estudos Tecnológicos, 3(1), 38-50.

[23] Stolfo, S. J., Fan, W., Lee, W., Prodromidis, A., Chan, P. K.(2000). Cost-based modeling for fraud and intrusion detection:Results from the JAM project. COLUMBIA UNIV NEWYORK DEPT OF COMPUTER SCIENCE.

[24] de Campos Souza, P. V., Silva, G. R. L., Torres, L. C. B. (2018,May). Uninorm based regularized fuzzy neural networks. In2018 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS) (pp. 1-8). IEEE.

[25] de Campos Souza, P. V., de Oliveira, P. F. A. (2018, April).Regularized fuzzy neural networks based on nullneurons for problems of classification of patterns. In 2018 IEEE Symposium on Computer Applications Industrial Electronics (ISCAIE) (pp. 25-30). IEEE.

[26] Huang, G. B., Zhu, Q. Y., Siew, C. K. (2006). Extreme learning machine: theory and applications. Neurocomputing, 70(1-3)489-501.

[27] Maas, A. L., Hannun, A. Y., Ng, A. Y. (2013, June). Rectifier non linearities improve neural network acoustic models. In Proc.icml (Vol. 30, No. 1, p. 3).

[28] Lemos, A. P., Caminhas, W., Gomide, F. (2012, August). A fast learning algorithm for uninorm-based fuzzy neural networks. In Fuzzy Information Processing Society (NAFIPS), 2012 Annual Meeting of the North American (pp. 1-6). IEEE.

[29] Friedman, J., Hastie, T., Tibshirani, R. (2001) The elements of statistical learning (Vol. 1, No. 10). New York, NY, USA::Springer series in statistics.

[30] Efron, B., Hastie, T., Johnstone, I., Tibshirani, R. (2004). Leastangle regression. The Annals of statistics, 32(2), 407-499.

[31] Girosi, F., Jones, M., Poggio, T. (1995). Regularization theory and neural networks architectures. Neural computation, 7(2), 219-269[32] Robert, C. (2014). Machine learning, a probabilistic perspective

[33] Hastie, T., Tibshirani, R.,Friedman, J. (2009). Unsupervised learning. In The elements of statistical learning (pp. 485-585). Springer, New York, NY.

[34] Bach, F. R. (2008, July). Bolasso: model consistent lasso estimation through the bootstrap. In Proceedings of the 25th international conference on Machine learning (pp. 33-40).ACM