

**FACULDADE DE TECNOLOGIA DE SÃO JOSÉ DOS CAMPOS
FATEC PROFESSOR JESSEN VIDAL**

LUIZ CARLOS FARIAS DA SILVA

**CHATLISTEN - UM SISTEMA PARA DETECÇÃO DE
ASSEDIAADORES SEXUAIS NO CHAT DO FACEBOOK**

São José dos Campos
2015

LUIZ CARLOS FARIAS DA SILVA

**CHATLISTEN - UM SISTEMA PARA DETECÇÃO DE
ASSEDIAADORES SEXUAIS NO CHAT DO FACEBOOK**

Trabalho de Graduação apresentado
à Faculdade de Tecnologia de São
José dos Campos, como parte dos
requisitos necessários para a
obtenção do título de Tecnólogo em
Banco de Dados.

Orientador: Professor Mestre Emanuel Mineda Carneiro

São José dos Campos
2015

Dados Internacionais de Catalogação-na-Publicação (CIP)
Divisão de Informação e Documentação

FARIAS DA SILVA, Luiz Carlos.
Chatlisten - Um Sistema para Detecção de Assediadores Sexuais no Chat do Facebook.
São José dos Campos, 2015.
78f.

Trabalho de Graduação – Curso de Tecnologia em Banco de Dados,
FATEC de São José dos Campos: Professor Jessen Vidal, 2015.
Orientador: Professor Mestre Emanuel Mineda Carneiro.

1. Áreas de conhecimento. I. Faculdade de Tecnologia. FATEC de São José dos Campos: Professor Jessen Vidal. Divisão de Informação e Documentação. II. Título

REFERÊNCIA BIBLIOGRÁFICA –

FARIAS DA SILVA, Luiz Carlos. **Chatlisten - Um Sistema para Detecção de Assediadores Sexuais no Chat do Facebook**. 2015. 78f. Trabalho de Graduação - FATEC de São José dos Campos: Professor Jessen Vidal.

CESSÃO DE DIREITOS –

NOME DO AUTOR: Luiz Carlos Farias da Silva.

TÍTULO DO TRABALHO: Chatlisten - Um Sistema para Detecção de Assediadores Sexuais no Chat do Facebook.

TIPO DO TRABALHO/ANO: Trabalho de Graduação / 2015.

É concedida à FATEC de São José dos Campos: Professor Jessen Vidal permissão para reproduzir cópias deste Trabalho e para emprestar ou vender cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte deste Trabalho pode ser reproduzida sem a autorização do autor.



Luiz Carlos Farias da Silva
Rua H30D, 124 - Campus do CTA
CEP 12228-820 – São José dos Campos – SP

LUIZ CARLOS FARIAS DA SILVA

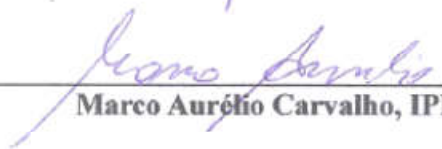
**CHATLISTEN - UM SISTEMA PARA DETECÇÃO DE
ASSEDIADORES SEXUAIS NO CHAT DO FACEBOOK**

Trabalho de Graduação apresentado
à Faculdade de Tecnologia de São
José dos Campos, como parte dos
requisitos necessários para a
obtenção do título de Tecnólogo em
Banco de Dados em 2015.

Composição da Banca



Helivelton Ferreira, Mestre, FATEC



Marco Aurélio Carvalho, IPEV



Emanuel Mineda Carneiro, Mestre, FATEC

15 / 12 / 2015

DATA DA APROVAÇÃO

À minha esposa Patrícia, aos meus
filhos Júlia, Laura e Pedro, à minha
mãe Tereza de Jesus e ao meu irmão
Javan.

AGRADECIMENTOS

Agradeço ao professor e orientador Emanuel Mineda Carneiro, por todo apoio, paciência e encorajamento dispensados continuamente ao longo de praticamente toda a graduação.

Agradeço à minha esposa Patrícia e aos meus filhos Júlia Maria, Laura Maria e Pedro, por todo apoio, dedicação e amor.

Agradeço ao meu irmão Javan pelos aconselhamentos e por acreditar sempre no meu sucesso.

Agradeço imensamente a todos os companheiros de sala de aula que trilharam os mesmos duros caminhos desta jornada, sempre oferecendo alguma forma de apoio e incentivo.

A todos os Professores agradeço profundamente por todos os ensinamentos passados e pelas experiências compartilhadas.

Por fim, agradeço a Deus, fonte de tudo e todos, por ter permitido que eu chegasse até aqui.

“O preço da liberdade é a eterna vigilância.”

John Philpot Curran

RESUMO

O assédio sexual contra menores na Internet vem crescendo de modo alarmante. Este trabalho tem por objetivo a criação de um sistema de monitoramento do mecanismo de chat do Facebook visando a prevenção do assédio sexual contra menores. Para a consecução deste trabalho foram usadas e integradas diversas ferramentas para coletar, tratar, analisar e classificar todo o texto capturado durante um bate-papo no Facebook. Dentre as técnicas utilizadas, destacam-se linguística computacional, processamento de linguagem natural, redes neurais artificiais e mineração de textos. Este trabalho permite monitorar um usuário conectado no sistema de chat do Facebook e atribuir uma pontuação referente à possibilidade do interlocutor se tratar de um predador sexual. O presente trabalho permite também a vigilância de crianças e adolescentes usando o sistema de chat do Facebook.

Palavras-Chave: predador sexual; assédio sexual; bate-papo; mineração de textos; inteligência artificial; processamento de linguagem natural.

ABSTRACT

Sexual harassment of minors on the Internet has been increasing in a very fast rate. This paper aims to create a monitoring mechanism for the Facebook chat system to prevent sexual harassment of minors. In order to accomplish this objective some tools were used and integrated to collect, treat, analyze, and classify text captured during a chat session. Among all the techniques applied in this work, it is important to highlight computational linguistics, natural language processing, artificial neural networks, and text mining. This work provides a tool for monitoring a user connected to a chat system along with a score indicating the possibility of having a sexual predator as a chat partner. This work also provides surveillance of children and teenagers using Facebook chat systems.

Keywords: sexual predator; sexual harassment; chat; text mining; artificial intelligence; natural language processing.

LISTA DE FIGURAS

Figura 1- As 10 Redes Sociais mais Acessadas no Brasil	17
Figura 2- Cadastro no Facebook	18
Figura 3- Redes Sociais no Brasil	19
Figura 4- Página Inicial do PJ	22
Figura 5 - Trecho de chat extraído do PJ	23
Figura 6 – Aplicações de Aprendizado de Máquina	25
Figura 7 - Neurônio Biológico X Neurônio Artificial	26
Figura 8- Sistemas Especialistas e a IA	29
Figura 9 - Etapas de KDD	31
Figura 10 – Modelo Geral de Processamento de Linguagem Natural	33
Figura 11 - Arquiteturas básicas cliente-servidor web, e-mail e XMPP	37
Figura 12 – Sequenciamento - somente agressor com modelo HMM	39
Figura 13 – ChatCoder com Logs do PJ	40
Figura 14 – Sequência Geral do Processo	41
Figura 15- Bibliotecas do Script1	43
Figura 16- Trecho de Download dos Logs do PJ do Script1	44
Figura 17- Limpeza dos Logs	45
Figura 18 - Limpeza nos Logs da NPS	45
Figura 19 – Classificação - POStagging	46
Figura 20 - Diagrama de Classes – Etapa de Transformação	47
Figura 21 – Sequência de Treinamento e Validação	50
Figura 22 – Treinamento / Testes	51
Figura 23 – Treinamento e Validação	52
Figura 24 – Sequência de Avaliação / Checagem	53
Figura 25 – Matrizes de Confusão	54
Figura 26 – Código de Avaliação / Checagem	54
Figura 27 – Sinótico Geral do Sistema	56
Figura 28 – Diagrama Geral do ChatListen	57
Figura 29 – Sequência de Captura e Monitoração	58
Figura 30 – Detalhes da Sequência de Captura e Monitoração	59
Figura 31 – Captura das Mensagens	59
Figura 32 – Sequência do Cálculo da Pontuação	60

Figura 33 – Cálculo da Pontuação	61
Figura 34 – Tela do Sistema ChatListen	62
Figura 35 – Pontuação com Sistema Completo e Log do PJ	64
Figura 36 – Alarme Visual com Log do PJ	65
Figura 37 – Alarme Visual com Log do CyberSex	65
Figura 38 – Alarme por Palavra Específica	66
Figura 39 – Resultado dos Testes	68

LISTA DE TABELAS

Tabela 1- Soluções e Pesquisas Existentes	42
Tabela 2- Descrição do Vetor de Entrada da RNA	48
Tabela 3- Dicionário de Dados das Tabelas de Vetores	49
Tabela 4 – Ajustes da RNA para Treinamento e Validação	52
Tabela 5 – Teste com <i>Logs</i> do PJ	66
Tabela 6 – Teste com <i>Logs</i> da NPS	67
Tabela 7 - Teste com <i>Logs</i> do CyberSex	67
Tabela 8 – Cruzamento Título / Características	69

LISTA DE ABREVIATURAS E SIGLAS

ANSI	<i>American National Standards Institute</i>
API	<i>Application Programming Interface</i>
CID	Código Internacional de Doenças
CMC	Comunicação Mediada por Computador
DML	<i>Data Manipulation Language</i>
IDE	<i>Integrated Development Environment</i>
JPA	<i>Java Persistency API</i>
MLP	<i>Multi Layer Perceptron</i>
NLTK	<i>Natural Language ToolKit</i>
NPS	<i>Naval Postgraduate School</i>
OpenNLP	<i>Open Natural Language Processing</i>
ORM	<i>Object-Relational Mapping</i>
PAN	<i>Plagiarism Authorship and Misuse of Software</i>
PJ	<i>Perverted-Justice</i>
PJFI	<i>Perverted Justice Foundation Incorporated</i>
PLN	Processamento de Linguagem Natural
POM	<i>Project Object Model</i>
POSTagger	<i>Part-Of-Speech Tagger</i>
RNA	Rede Neural Artificial
SGBD	Sistema de Gerenciamento de Bases de Dados
SMS	<i>Short Messaging Service</i>
SQL	<i>Structured Query Language</i>
SVM	<i>Software Vector Machine</i>
SWN	<i>SentiWordNet</i>
TLS	<i>Transport Layer Security</i>
XML	<i>eXtensible Markup Language</i>
XMPP	<i>eXtensible Messaging and Presence Protocol</i>

SUMÁRIO

1- INTRODUÇÃO	16
1.1- Contextualização	16
1.2- Motivação	17
1.3- Objetivos do Trabalho	19
1.4- Escopo do Trabalho	19
1.5- Especificação de Requisitos	20
1.6- Conteúdo do Trabalho	20
2- FUNDAMENTAÇÃO TEÓRICA	21
2.1- Redes Sociais na Internet	21
2.2- Assédio Sexual Contra Menores	21
2.3- Perverted-Justice (PJ)	22
2.4- Inteligência Artificial – (IA)	23
2.4.1- Aprendizado de Máquina (AM)	24
2.4.2- Técnicas	25
2.5- <i>KnowledgeDiscoveryinDatabases</i> (KDD)	30
2.5.1- Seleção - <i>Selection</i>	31
2.5.2- Pré-processamento - <i>Preprocessing</i>	31
2.5.3- Transformação - <i>Transformation</i>	31
2.5.4- Mineração de Dados - <i>Data Mining</i>	31
2.5.5- Interpretação / Validação – <i>Interpretation / Evaluation</i>	32
2.6- Processamento de Linguagem Natural - PLN	32
2.6.1- Linguística Computacional - LC	33
2.6.2- Método de Bogdanova (2012)	34
2.7. Ferramentas Utilizadas	34
2.7.1- Editor Notepad++ v6.8.6	35
2.7.2- Python IDLE 2.7.10	35
2.7.3- SentiWordNet 3.0	35
2.7.4- WordNet-Affect – Strapparava (2004)	35
2.7.5- IDE Eclipse Java Versão Luna 2(4.4.2)	36
2.7.6- Apache Maven 3.3.3	36
2.7.7- Biblioteca JavaMail 1.5.3	36

2.7.8-	Biblioteca EclipseLink 2.5.2	36
2.7.9-	JavaFX 2.0	36
2.7.10-	Scene Builder 2.0	37
2.7.11-	Biblioteca Smack versão 4.1.1	37
2.7.12-	MySQL Server 5.6 e MySQL Workbench 5.6 CE	38
2.7.13-	OpenNLP 1.6.0	38
2.8.	Soluções e Pesquisas Existentes	38
3-	DESENVOLVIMENTO	43
3.1-	Arquitetura de Treinamento	43
3.1.1-	Seleção (<i>Download</i>)	43
3.1.2-	Extração / Limpeza	44
3.1.3-	Transformação	45
3.1.4-	Mineração dos Dados – Busca por Padrões	49
3.1.5-	Avaliação / Checagem	53
3.2-	Arquitetura de Detecção	55
3.2.1-	Captura das Mensagens do Facebook	56
3.2.2-	Cálculo da Pontuação	59
4-	ANÁLISE E DISCUSSÃO DOS RESULTADOS	63
4.1-	Teste de Detecção	63
4.2-	Análise dos Resultados	66
4.3-	Comparativo entre Soluções Existentes e Sistema Proposto	68
5-	CONCLUSÃO	70
5.1-	Contribuições	70
	Pesquisa Bibliográfica	70
	Monitoração do chat do Facebook	70
	Cálculo de Pontuação em chatse Análise de Sentimentos	70
	Monitoração e Vigilância de Crianças e Adolescentes	71
	Testes com RNA Usando Vetores Gerados a partir de Textos	71
5.2-	Conclusão Geral	71
5.3-	Sugestões para Trabalhos Futuros	71
	Elevar a Quantidade de <i>Logs</i> de Não Predadores	71
	Aprimorar os Métodos de Teste	71
	Integrar Alarmes via SMS	72
	Permitir Escolha de Diferentes Métodos de Classificação	72

Adaptar ou Criar Sistemas para o Português Brasileiro	72
REFERÊNCIAS	73

1- INTRODUÇÃO

Este capítulo apresenta a contextualização da pesquisa, a sua motivação e os seus objetivos. Ele também define o escopo da pesquisa e sua especificação de requisitos. Por fim, ele apresenta o conteúdo do trabalho.

1.1- Contextualização

O surgimento dos *sites* de compartilhamento e das redes sociais na Internet facilitou e acelerou a interação entre as pessoas. Conforme Moura (2007) a comunicação mediada por computador (CMC) foi amplamente potencializada com o surgimento da Internet. A Internet fornece benefícios como, por exemplo, o amplo acesso à informação e à pesquisa.

Juntamente com os avanços da Internet surgem novos modos de interação social na forma de aplicativos ou dispositivos *desktop* ou *mobile* que permitem às pessoas conectarem-se e trocarem informações em tempo real, em qualquer lugar e anonimamente. Nesse contexto de comunicações velozes, sem fronteiras e anônimas Sydow (2009) vê que a Internet, assim como o mundo real, é um ambiente de risco, mas potencializado.

Não há como citar o assédio sexual contra menores sem mencionar a pedofilia, definida, conforme classificação CID10 F65.4 (Código Internacional de Doenças – subgrupo F65.4):

“Um transtorno da preferência sexual caracterizado pela preferência sexual, quer se trate de meninos, meninas ou de crianças de um ou do outro sexo, geralmente pré-púberes ou no início da puberdade.”

O termo pedófilo não é caracterizado por um significado único e estanque, pois segundo Da Silva (2013), existem variações na terminologia conforme o tipo particular de conduta do agente. Assim, um novo conceito surge e será enfocado neste trabalho de graduação, o predador sexual.

Conforme Salter (2009), predador sexual é o indivíduo que age com cuidadosa premeditação e usa sofisticadas técnicas de enganação para ganhar a confiança das presas sexuais e, muitas vezes, desempenha papéis duplos na comunidade.

De acordo com Moreira (2008) o exercício da sexualidade na atualidade começa cada vez mais cedo e é impulsionado pela imposição social que leva as crianças a tornarem-se adolescentes e os adolescentes a ingressarem na vida adulta precocemente. Assim, ainda não psicologicamente preparados para *adolescere* e

ingressarem na vida adulta, as crianças e os jovens tornam-se alvos fáceis e muitas vezes desprotegidos para predadores sexuais.

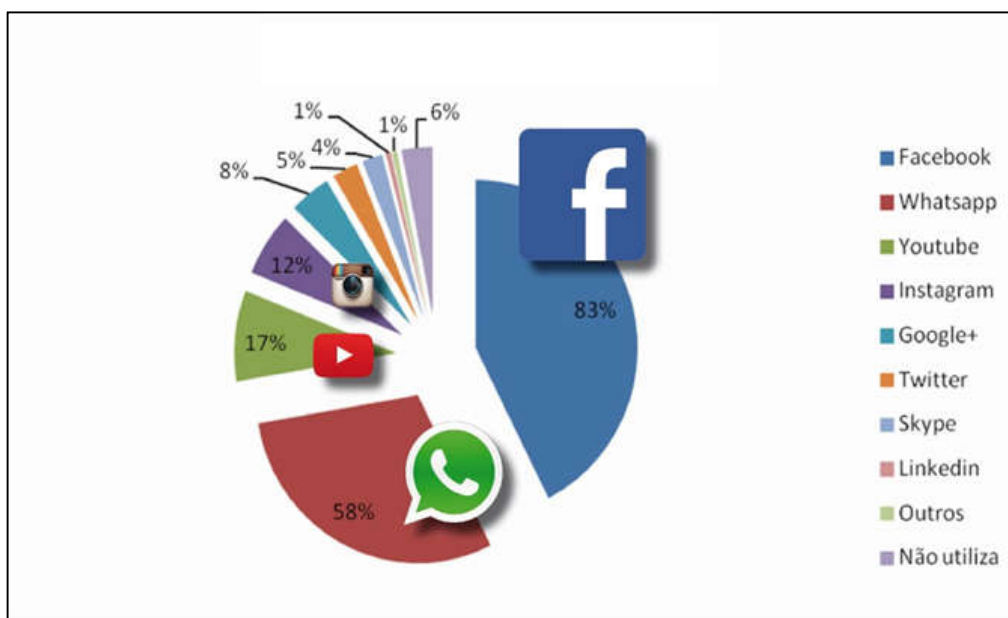
Com relação ao entendimento sobre assédio sexual, tem-se ainda uma primeira noção de que somente existe assédio quando há relações trabalhistas envolvidas, porém, conforme novos entendimentos doutrinários e segundo Pamplona Filho (2002), o assédio sexual, sob uma nova ótica, configura-se quando coexistem os seguintes elementos: sujeito agente, o assediador; sujeito destinatário, o assediado; conduta de natureza sexual; rejeição à conduta do agente; e reiteração da conduta.

1.2- Motivação

Existe uma quantidade razoável de sites de redes sociais disponíveis na Internet. A Figura 1 apresenta um gráfico contendo as dez redes sociais¹ mais acessadas no Brasil conforme Pesquisa Brasileira de Mídia, (PBM, 2015).

Nota-se que o Facebook lidera o *ranking* de acesso às redes sociais da Internet, com 83%. O mecanismo de troca de mensagens eletrônicas mais usado é o Whatsapp – serviço de troca de mensagens eletrônicas adquirido em outubro de 2014 pelo Facebook – com 58%.

Figura 1- As 10 Redes Sociais mais Acessadas no Brasil



Fonte: PBM (2015)

¹ No gráfico da Figura 1, as fatias representam o percentual de acessos de cada rede social comparado com o acesso às redes de mesmo tipo. Não são fatias do mesmo gráfico tipo pizza.

Com relação à rede social enfocada neste trabalho de graduação, o Facebook, o processo de cadastro exige somente uma conta de e-mail válida para que se consiga acessar a rede social e, a partir daí, começar a usar suas facilidades. Na Figura 2, são apresentadas instruções tiradas da página de ajuda do Facebook sobre como efetuar o cadastro na referida rede social.

Conforme Gonzaga (2014) praticamente toda criança brasileira com idade entre seis e nove anos filha de pai ou mãe que usa a Internet já está conectada e metade destas crianças está no Facebook.

Figura 2- Cadastro no Facebook

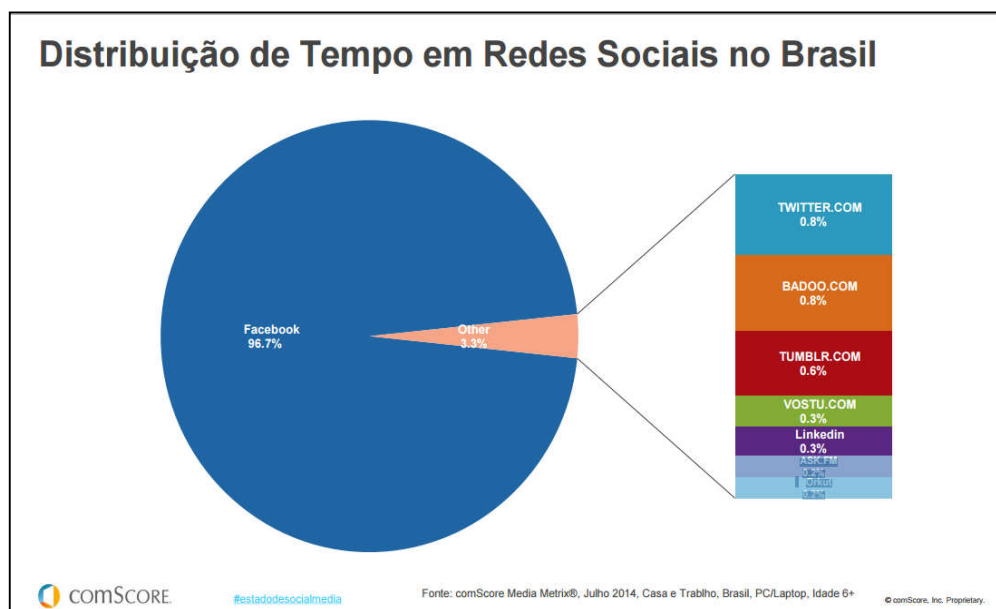


Fonte: Facebook

Para que se possa ter uma noção da hegemonia do Facebook com relação às demais redes sociais, a Figura 3 apresenta a distribuição do tempo gasto em redes sociais no Brasil no ano de 2014. Nota-se que o tempo gasto pelos usuários das demais redes não totaliza 4%.

Com base nestes dados, o presente trabalho de graduação tem como foco o mecanismo de chat do Facebook.

Figura 3- Redes Sociais no Brasil



Fonte: comScore

A principal motivação para a criação da ferramenta apresentada neste trabalho de graduação reside na falta de mecanismos e sistemas eficazes que permitam não somente monitorar e acompanhar as conversas ocorridas nos sites de relacionamento usando a Internet como meio.

A ferramenta proposta deve também ser eficaz no reconhecimento automático de padrões de conduta imprópria que possam desdobrar-se em assédio sexual, haja vista o alarmante e crescente número de casos (JACOBS, 2012) (BASHER, 2014).

1.3- Objetivos do Trabalho

Este trabalho tem por objetivo criar e implementar um sistema para monitoração em tempo real do mecanismo de chat do Facebook com vistas à proteção de menores de idade por intermédio de identificação de predadores sexuais.

1.4- Escopo do Trabalho

Apesar de ser de extrema urgência e relevância para a sociedade o combate contra quaisquer formas de assédio, neste trabalho de graduação, será focado apenas o assédio sexual contra menores.

Este Trabalho de Graduação restringe-se à criação e implementação de um aplicativo para desktops para monitoração do mecanismo de chat do Facebook e

atribuir automaticamente uma pontuação para o conteúdo do chat conforme a possibilidade de haver ou não um predador sexual usando o referido chat.

1.5- Especificação de Requisitos

O presente trabalho de graduação propicia:

- Uma revisão bibliográfica adequada;
- A Concepção e implementação de um sistema para monitoramento em tempo real do mecanismo de chat do Facebook visando identificação de predadores sexuais;
- Realização de testes de validação;
- Análise dos resultados obtidos; e
- Conclusões e sugestões para trabalhos futuros.

1.6- Conteúdo do Trabalho

Este trabalho de graduação está estruturado em cinco capítulos, cujo conteúdo é apresentado a seguir resumidamente.

No Capítulo 1 é feita a introdução e apresentação do trabalho com a contextualização, motivação, objetivos, escopo e requisitos. O Capítulo 2 apresenta a fundamentação teórica do trabalho com a conceituação das áreas abordadas. São também apresentadas as ferramentas propostas, bem como as soluções já existentes.

No Capítulo 3 tem-se detalhado todo o desenvolvimento do trabalho proposto. No Capítulo 4 são apresentadas algumas análises sobre os resultados alcançados, discutidas com base nos requisitos levantados e objetivos propostos.

Por fim, no Capítulo 5, têm-se as conclusões sobre todo o trabalho e as contribuições obtidas são apresentadas, bem como algumas sugestões para trabalhos futuros.

2- FUNDAMENTAÇÃO TEÓRICA

Neste Capítulo é apresentado o site *pervverted-justice.com* e são abordados os seguintes temas que servem de alicerce para o desenvolvimento do trabalho proposto: redes sociais na Internet, assédio sexual contra menores, processamento de linguagem natural (PLN), linguística computacional (LC), sistemas especialistas (SE), redes neurais artificiais (RNA), aprendizado de máquina (AM) e inteligência artificial (IA).

2.1- Redes Sociais na Internet

De acordo com Aguiar (2007) pode-se conceituar o termo redes como associações encadeadas, interações ou vínculos não hierarquizados envolvendo relações de comunicação com intercâmbio de informações e trocas culturais. Em termos históricos, os grupos de pessoas começaram a ser estudados como uma rede social a partir da década 1940 por antropólogos e psicólogos.

As redes sociais na Internet, da forma que se apresentam atualmente, estão fazendo parte da vida das pessoas cada vez mais. Por esse motivo novas áreas de estudo são criadas, fato que se confirma lendo o seguinte trecho conforme Aguiar (2007):

“... é nesse sentido que podem ser de grande valia as contribuições da Ciberantropologia, uma subárea da Antropologia Cultural que vem dedicando especial atenção ao ciberespaço como um “campo”, isto é, como um “espaço” interativo de relações socioculturais gerado pela comunicação mediada por computador (CMC).“

Percebe-se assim que as redes sociais na Internet estão presentes na vida de grande parte das pessoas e, por esta ótica, este trabalho de graduação tem, por campo de atuação da ferramenta proposta, a área do ciberespaço delimitada pelas redes sociais.

2.2- Assédio Sexual Contra Menores

Conforme o ordenamento jurídico brasileiro, o assédio sexual cuja descrição encontra-se no Código Penal Brasileiro (artigo 216-A, caput) fica caracterizado somente quando há relações de trabalho envolvidas. No entanto, Pamplona filho (2002) caracteriza a conduta de assédio sexual não somente como resultado das relações que ocorrem no ambiente de trabalho.

Nesse sentido, caso a Justiça e o Direito formassem um sistema estanque e não evolutivo, não haveria como caracterizar criminalmente as condutas praticadas pelos

predadores na Internet. Segundo Cavaliere Filho (2010) o Direito, como resultado da evolução social, adapta-se à situação atual criando mecanismos de composição e resolução de conflitos.

2.3- Perverted-Justice (PJ)

O PJFI (Perverted Justice Foundation Incorporated) é uma fundação estadunidense sem fins lucrativos que forma um sistema especializado em detectar assediadores sexuais por meio de técnicas próprias que envolvem recrutamento e treinamento de pessoal voluntário com a finalidade de se passarem por crianças ou adolescentes, coletando e registrando as conversas com possíveis predadores.

A referida fundação existe desde 2003 e trabalha de forma conjunta com a mídia e a polícia locais. Além de trabalhar na prevenção de crimes e na divulgação dos criminosos, a PJFI vem formando uma sólida base para estudos no que tange ao conhecimento e *modus operandi* dos predadores atuais.

Na Figura 4, vê-se o site do PJ podendo-se notar o aspecto de prevenção que o site apresenta, pois a cada atualização da página inicial, a foto de um predador condenado com a ajuda da PJFI é carregada. Os arquivos contendo as conversações de todos os predadores condenados encontram-se também disponíveis publicamente.

Figura 4- Página Inicial do PJ



Fonte: PJ

A Figura 5 apresenta um trecho de uma conversa retirada de um arquivo disponibilizado no PJ. O voluntário se passa por uma garota de 13 anos de idade com *nickname* `sadlilgrrl` e o predador condenado com a ajuda do PJ tem o *nickname* `fleet_captain_jaime_wolfe`. Entre parênteses e em letras azuis estão comentários adicionados pelo pessoal do PJ.

Figura 5 - Trecho de chat extraído do PJ

The predator in the chat-log you're about to read is named **Paul Short**. He was convicted in Illinois in regards to the following chat-log. He has since gone non-compliant after being released from prison. Short has ties to the Arizona area where his family lives and obviously has made residence in Illinois as well.

If you have any information regarding this dangerous internet predator and his current location, please contact the Illinois State Police immediately.

You won't see much commentary on this log. It essentially speaks for itself.

Saturday, March 20

First line was: `fleet_captain_jaime_wolfe` (1:32:26 PM): Why are you sad?

`sadlilgrrl` (1:33:04 PM): eh. i moved and dont know anyone and i have 2 be home schooled til august.

`fleet_captain_jaime_wolfe` (1:33:34 PM): Too bad. But, at least you are in school...

`sadlilgrrl` (1:33:50 PM): i guess so.

`fleet_captain_jaime_wolfe` (1:34:11 PM): So is that what is making you sad?

`sadlilgrrl` (1:34:29 PM): i just don't have any friends here and we moved here because my mom and dad split up.

`fleet_captain_jaime_wolfe` (1:34:34 PM): Where are you?

`sadlilgrrl` (1:34:37 PM): so mom's always at work or teaching me.

`sadlilgrrl` (1:34:43 PM): chicago area

`fleet_captain_jaime_wolfe` (1:34:52 PM): Well, at least your mother is taking an interest.

`fleet_captain_jaime_wolfe` (1:34:55 PM): Same here..

`sadlilgrrl` (1:35:07 PM): hah. we fight so much.

`fleet_captain_jaime_wolfe` (1:35:18 PM): What do you mean, you fight?

`sadlilgrrl` (1:35:47 PM): she just really doesn't understand that i miss my friends back home

`fleet_captain_jaime_wolfe` (1:35:56 PM): Oh? Well, how old are you? *(Asked.)*

`sadlilgrrl` (1:36:16 PM): i'm almost 14.*(Answered.)*

`fleet_captain_jaime_wolfe` (1:36:29 PM): I see... And, you want your mother to treat you like an adult? *(This man is an amazing groomer.)*

Fonte: PJ

2.4- Inteligência Artificial – (IA)

Os dicionários da língua portuguesa definem o termo inteligência como sendo a faculdade de aprender, compreender e adaptar-se, descartando outras definições de cunho teológico, sociológico e filosófico. É difícil uma definição estanque que esgote todo o significado do termo.

De acordo com Do Lago Pereira (2008), não há consenso sobre o significado de inteligência e, dessa forma, definir precisamente inteligência artificial é uma tarefa, se não impossível, pelo menos extremamente difícil.

Abandonando uma definição precisa, uniforme e final, Allan Turing em 1950 criou um teste com o qual, por meio de um terminal, um ser humano deveria interrogar “alguém” num local remoto similarmente ao que ocorre quando se está em bate-papo na Internet e, se após um determinado tempo, o ser humano não fosse capaz de perceber que esse “alguém” era uma máquina, a hipótese da existência de inteligência artificial estaria confirmada.

Passar pelo teste de Turing é extremamente difícil, pois envolve processamento de linguagem de máquina, representação de conhecimento, raciocínio automatizado e aprendizado de máquina. No entanto, apesar da extrema complexidade exigida para passar no teste de Turing, em junho de 2014, durante um teste realizado na Universidade de Reading, um programa de computador do tipo *chatterbot*² nomeado Eugene Goostman conseguiu passar no referido teste.

Existem duas abordagens diferentes sobre IA – IA forte e IA fraca. Conforme Souza (2014) a IA forte baseia-se na capacidade do computador em resolver problemas por meio do raciocínio e da lógica e a IA fraca baseia-se na solução de problemas não determinísticos, estando entre o processamento e reconhecimento com uma linguagem natural.

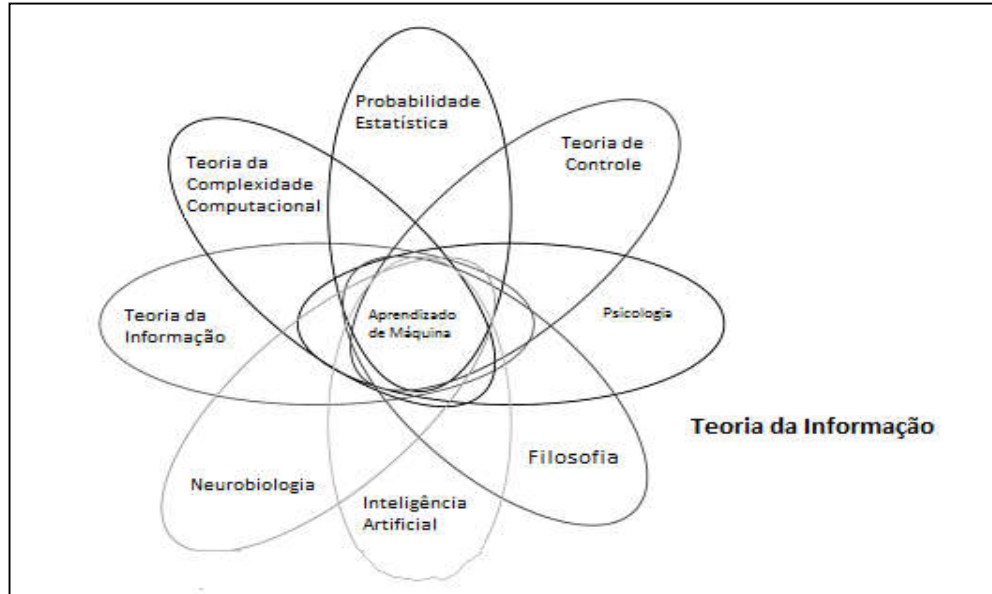
2.4.1- Aprendizado de Máquina (AM)

Conforme Monard (2003), aprendizado de máquina é uma área de IA cujo objetivo é o desenvolvimento de técnicas computacionais sobre o aprendizado bem como a construção de sistemas capazes de adquirir conhecimento de forma automática.

De acordo com Carbonell (1983), em uma tradução aberta, aprendizado de máquina refere-se ao estudo e modelagem computacional de processos de aprendizado em suas múltiplas manifestações. As referidas manifestações podem ser visualizadas na Figura 6.

² *Chatterbots* são aplicativos que simulam uma conversa de um ser humano. Conceito disponível em: <http://www.ies.ufpb.br/ojs/index.php/ies/article/view/1758>.

Figura 6 – Aplicações de Aprendizado de Máquina



Fonte: Adaptada de Kapitanova (2012)

Outra definição em tradução aberta e de acordo com Kapitanova (2013) refere-se ao aprendizado de máquina como o projeto e desenvolvimento de algoritmos que permitam sistemas a usarem dados empíricos, experiência e treinamento para desenvolverem-se e adaptarem-se às mudanças que ocorrem no meio ambiente.

2.4.2- Técnicas

Aprendizado de máquina propõe métodos ou técnicas que auxiliam a criação de sistemas onde existe a necessidade de automatização de tarefas que em geral levam muito tempo para responder ou quando não é possível criar um modelo matemático para resolver um determinado problema.

Existem diversas técnicas de AM e o que determina qual técnica é aplicável a qual problema depende de vários fatores, porém o fator preponderante é o tempo.

2.4.2.1- Redes Neurais Artificiais – RNA

Conforme Thomé (2002) as redes neurais ou redes neuronais são modelos computacionais que emulam a estrutura e o funcionamento do cérebro humano. Por meio de uma rede de neurônios artificiais – elementos de processamento muito simples

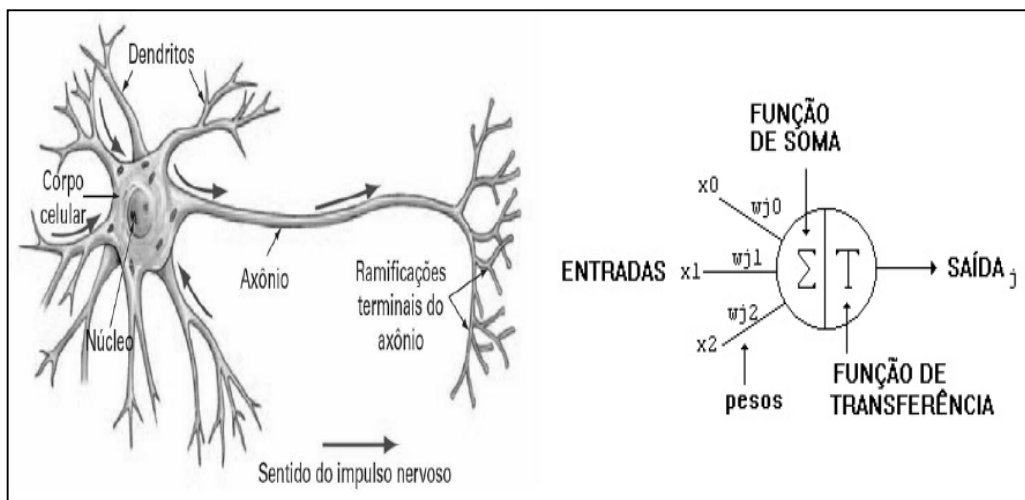
– altamente interconectada e paralela as RNA são capazes de reconhecer padrões e realizar a aproximação de funções.

A Figura 7 apresenta uma comparação entre os neurônios biológicos e artificiais. De forma semelhante ao neurônio biológico, o neurônio artificial recebe estímulos (entradas) que, depois de processadas, geram um impulso (saída).

Os dendritos são os elementos receptores, a entrada do neurônio. O axônio é a linha de transmissão que transporta o sinal de saída do neurônio. As sinapses são a região onde a saída de um neurônio e a entrada de outro se conectam. O corpo celular é a parte do neurônio responsável pelo “processamento” dos sinais de entrada. Quando os valores das entradas atingem um determinado limiar, o neurônio libera um impulso elétrico para o axônio.

Por se tratar de uma rede, os impulsos (saídas) tornam-se estímulos (entradas) de outras células e o comportamento inteligente destas redes vem justamente das interações que ocorrem entre as unidades da rede – saída de uma unidade é uma das entradas de outra. Assim, com uma saída conhecida, as várias ramificações interagem para encontrar valores de modo que a saída produzida seja o mais próximo possível da saída conhecida.

Figura 7 - Neurônio Biológico X Neurônio Artificial



Fonte: Franciscani (2012)

As redes neurais têm as seguintes características:

- São modelos adaptativos treináveis;
- Podem representar domínios complexos (não lineares);

- São capazes de generalização diante de informações incompletas;
- São capazes de armazenar informações de modo associativo;
- Possuem alto paralelismo e por isso são muito velozes; e
- Processam informações espaço/temporais.

Thomé (2002) explica que em uma RNA não existe a ideia de programa, onde o programador cria e codifica uma estratégia para solucionar um problema e que também não se tem a ideia de um conhecimento explícito e armazenado que conduza a busca por uma resolução do problema enfrentado, pois a rede é dinâmica, não possui memória (nos moldes biológicos que se conhece), não acessa nem possui arquivos de dados e não é programável.

Conforme Cardon (1994) pode-se distinguir as RNA por pelo menos dois componentes físicos: conexões e elementos de processamento. Pode-se fazer uma analogia com um grafo orientado, onde os nodos são os elementos de processamento e as arestas representam as conexões. A junção desses elementos forma a RNA.

Existem dois outros componentes não físicos: padrões e funções. Os padrões são as entradas do sistema e as funções são os modelos matemáticos utilizados no treinamento e reconhecimento de padrões (Simpson, 1990).

Estes quatro componentes (conexões, elementos de processamento, padrões e funções) formam o conjunto básico de qualquer RNA, porém essa denominação não é regra, visto que não existem normas que orientem o uso de cada componente.

Segundo Kohonen (1990) e Lippman (1987) existem diferenças marcantes entre as redes mesmo não havendo norma que defina seus modelos. Podem-se distinguir os modelos por meio de suas características básicas: tipo de entradas, forma de conexão e tipo de aprendizado.

Com relação às entradas, estas são de dois tipos: binárias e intervalares. As binárias aceitam elementos discretos na forma de 0 ou 1 ao passo que as intervalares aceitam qualquer valor numérico (forma contínua) como entrada.

As formas de conexão são de três tipos: alimentação à frente, retroalimentação e competitiva. Na forma de alimentação à frente, os valores de entrada são simplesmente transformados em valores de saída. Na retroalimentação os valores de entrada são alterados em diversas mudanças de estado, sendo que a saída também

alimenta a entrada. A forma competitiva realiza interação lateral dos valores recebidos na entrada entre os elementos no interior de uma zona de vizinhança.

O tipo de aprendizado refere-se à existência ou não de um valor de saída pré-estabelecido para a rede. Assim existem os tipos: supervisionado e autoaprendizado ou não supervisionado. No supervisionado existe uma predefinição para a saída e o não supervisionado não possui.

Com base nas características apresentadas e das relações existentes entre elas, pode-se classificar as RNA nos seguintes modelos:

- Perceptron ou Adaline – Tem entrada intervalar, aprendizado supervisionado e alimentação à frente;
- Retro propagação ou Backpropagation – Possui entrada intervalar, aprendizado supervisionado e alimentação à frente;
- Hofield – Tem entrada binária, aprendizado supervisionado e com retroalimentação; e
- Kohonen – Apresenta entrada intervalar, autoaprendizado e conexão competitiva.

Uma generalização do modelo Perceptron é o MLP (Multi Layer Perceptron) onde existem várias camadas de nodos (comparação com grafos) ou elementos de processamento entre a entrada e a saída.

De acordo com Castro (2010) as redes MLP têm sido aplicadas com sucesso em diversas áreas, executando várias tarefas como classificação (reconhecimento) de padrões e controle e processamento de sinais.

Neste trabalho de graduação será usado este tipo de rede neural artificial.

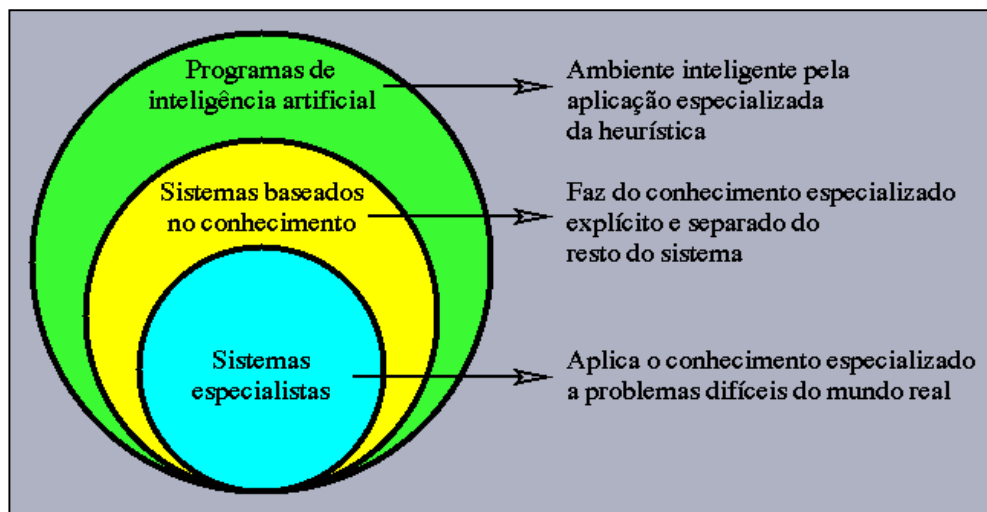
2.4.2.2- Sistemas Especialistas– SE

Segundo Grossmann Jr (apud SCHILDT, 1987) sistemas especialistas são programas de computador que procuram reproduzir o comportamento de um especialista humano em alguma área. Para que isto ocorra estes sistemas utilizam informações fornecidas pelo usuário para formar uma opinião sobre um determinado assunto. Assim o sistema especialista faz perguntas e o usuário responde, até que ele possa identificar uma hipótese que se adapte às respostas.

Os sistemas especialistas nada mais são do que sistemas que imitam a estratégia de resolução de problemas de um especialista humano e, apesar de obviamente não possuírem a capacidade cognitiva de um especialista humano, revelam-se como ferramentas extremamente importantes na resolução de problemas que exigem respostas rápidas.

Tem-se na Figura 8 os Sistemas Especialistas com relação à inteligência artificial (IA), podendo-se notar que os referidos sistemas são especialidades englobadas nos sistemas baseados no conhecimento que estão inseridos no conceito de programas de IA.

Figura 8- Sistemas Especialistas e a IA



Fonte: UFMA

2.4.2.3- Lógica Fuzzy - LF

Conforme Coppin (2010) a Lógica Fuzzy (Nebulosa) também conhecida como lógica multivalorada, diferentemente das lógicas que apresentam valores binários para

representar suas respostas, trabalha com valores polivalentes para os resultados, podendo apresentar valores intermediários entre um máximo e um mínimo.

Os valores intermediários não denotam valores verdadeiros ou falsos, mas dizem o quão próximos dos extremos os valores das respostas estão. Percebe-se, portanto que a LF trabalha com incertezas e inexatidões do conhecimento.

Como exemplo pode-se usar a altura das pessoas como valor de entrada. Um valor de 2,4 metros é considerado alto e um valor de entrada de 1,2 metro é considerado baixo. Um valor de 1,75 metro estaria, portanto, dentro de alguma faixa intermediária – alta, muito alta, baixa, muito baixa, etc.

A lógica Fuzzy tem aplicação nas áreas que envolvem tomadas de decisão quando as informações de entrada no sistema são imprecisas e qualitativas. Assim, de acordo com Rezende (2005) os modelos Fuzzy são especialmente adequados em processos que exigem tomadas de decisão por parte de operadores e agentes de operação.

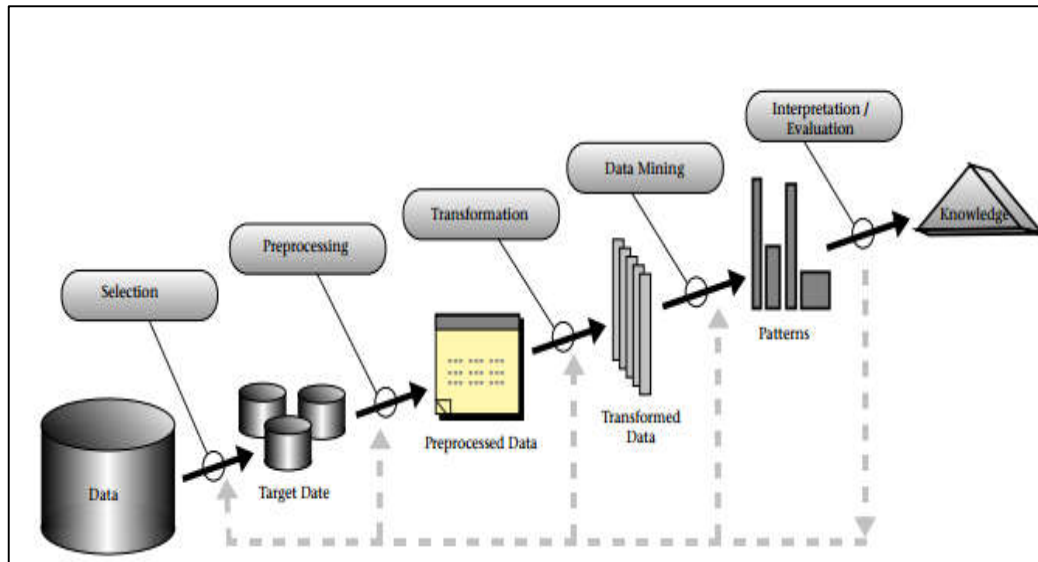
2.5- *Knowledge Discovery in Databases (KDD)*

A sigla KDD – *Knowledge Discovery in Databases* (descoberta de conhecimento em bases de dados) é usada para designar um processo abrangente de descoberta de conhecimento que tem a seguinte definição conforme Fayyad et al (2006):

“É um processo, de várias etapas, não trivial, interativo e iterativo, para identificação de padrões compreensíveis, válidos, novos e potencialmente úteis a partir de bases de dados.”

Na Figura 9 encontra-se a sequência do processo de KDD com cinco fases e os entes que entram e saem em cada fase. Tem-se a entrada geral que é a base de dados brutos e dessa base são selecionados os dados que formam os entes que entram na próxima etapa. Esse processo de adequação de dados para alimentar as próximas etapas por meio de técnicas próprias a cada fase é aplicado até que se tenha o conhecimento desejado.

Figura 9 - Etapas de KDD



Fonte: Fayyad (2006)

Nos subcapítulos 2.5.1 a 2.5.5 os processos de KDD são resumidos segundo Fayyad (2006).

2.5.1- Seleção - *Selection*

Nesta etapa, dentre os dados existentes, aqueles a serem usados são analisados e selecionados com a finalidade de se buscar por padrões para a geração de conhecimento novo e útil.

2.5.2- Pré-processamento - *Preprocessing*

Nesta fase os dados são preparados e tratados para que possam ser usados pelos algoritmos das fases posteriores. Deve-se nesta etapa de KDD identificar e remover os valores que se apresentam inválidos, repetidos ou inconsistentes – procede-se com a limpeza dos dados.

2.5.3- Transformação - *Transformation*

Em algumas ocasiões os dados precisam ser transformados para que possam ser usados. As transformações podem ser lineares ou não lineares de modo que os melhores ou mais relevantes valores sejam encontrados.

2.5.4- Mineração de Dados - *Data Mining*

Conforme Thome (2002) Mineração de dados é referida em alguns casos e por algumas comunidades específicas ainda de modo similar ao próprio processo geral de

busca por padrões ou oportunidades de conhecimento em conjuntos de dados brutos, confundindo assim com a própria definição de KDD.

Apesar da aparente confusão terminológica ainda presente, a mineração de dados pode ser resumidamente definida como a busca por padrões por meio de aplicação de algoritmos e técnicas computacionais específicas.

2.5.5- Interpretação / Validação – *Interpretation / Evaluation*

Nesta última etapa do processo de descoberta do conhecimento em bases de dados, procede-se com a análise dos resultados da mineração, permitindo assim a geração de conhecimento pela interpretação e utilização dos resultados.

2.6- Processamento de Linguagem Natural - PLN

Por linguagem natural entende-se a linguagem escrita e falada pelos seres humanos de forma que não haja a intervenção de quaisquer técnicas ou dispositivos, ou seja, é a linguagem que os seres humanos usam em suas interações.

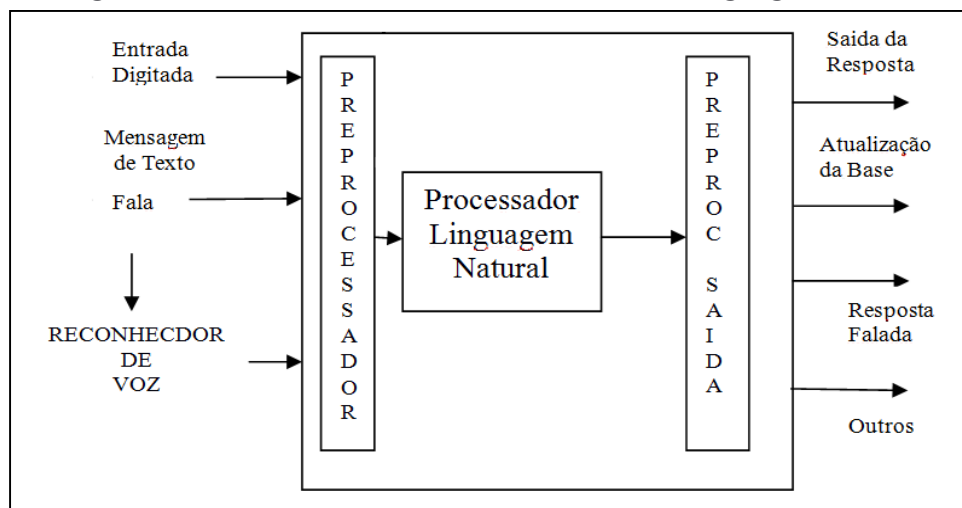
De acordo com Do Lago Pereira (2008), Processamento de Linguagem Natural (PLN) consiste no desenvolvimento de modelos computacionais para a realização de tarefas que dependem de informações expressas em uma língua natural. Uma definição mais abrangente de acordo com Chowdhury (2003) reside na seguinte tradução aberta: “é uma área de pesquisa e aplicação que explora como computadores podem ser usados para entender e manipular textos ou discursos em linguagem natural para fazerem coisas úteis”.

Pesquisadores de Processamento de Linguagem Natural tem o objetivo de reunir conhecimento sobre como seres humanos entendem e usam a linguagem de forma que ferramentas e técnicas apropriadas possam ser desenvolvidas para fazer sistemas computacionais entenderem e manipularem linguagens naturais para realizar as tarefas desejadas.

Um processador de linguagem natural visto de forma básica é apresentado na Figura 10. As entradas do sistema são as linguagens naturais de tipos escritos (textos prontos ou digitação) ou a própria fala. Antes do processamento propriamente dito, as entradas devem passar por uma etapa de adequação e transformação a fim de permitir a extração de significados úteis.

Por fim, tem-se a informação e, com base nesta, pode-se alimentar outros sistemas na cadeia, sendo estes, por exemplo: uma base de dados, uma saída simples de texto ou qualquer outra forma de saída que se faça necessária.

Figura 10 – Modelo Geral de Processamento de Linguagem Natural



Fonte Adaptada de BATES (1995)

2.6.1- Linguística Computacional - LC

Por Linguística Computacional tem-se a seguinte definição de acordo com VIEIRA (2001):

“É a área do conhecimento que explora as relações entre linguística e informática, tornando possível a construção de sistemas com capacidade de reconhecer e produzir informação apresentada em linguagem natural.”

No caso deste trabalho de graduação, a definição aplica-se à construção de um sistema capaz de reconhecer informação apresentada em linguagem natural e, mais especificamente ainda, a linguagem esbarra nos limites do selvagem de tão natural que se apresenta nos sites de bate-papo.

Outra definição mais abrangente para a Linguística Computacional é a seguinte:

“Uma área híbrida que envolve pesquisadores da linguística e da Informática. A Linguística Computacional é a parte da ciência linguística que se preocupa com o tratamento computacional da linguagem, e suas aplicações englobam programas como tradutores automáticos, *chatbots*, corretores ortográficos e gramaticais, *parsers*, entre outros. (DE ÁVILA OTHERO, 2006).”

Nesta segunda definição, nota-se a necessidade de cooperação entre a Linguística como Ciência própria e a informática também como Ciência, porém vista aqui tão somente como ferramenta.

Em resumo, a Linguística Computacional vale-se dos conhecimentos totais da Língua com a poderosa ajuda da computação informatizada.

2.6.2- Método de Bogdanova (2012)

No trabalho intitulado “*On the Impact of Sentiment and Emotion Based Features in Detecting On-line Sexual Predators*” de Bogdanova (2012), as variações emocionais marcantes e distintas que os predadores sexuais apresentam e que são perceptíveis nos textos formam um conjunto de características muito útil para a detecção com um grau considerável de acerto.

O método consiste, resumidamente, em detectar palavras específicas que denotam tipos de sentimentos e condutas que são marcantes na personalidade *estudada* dos pedófilos e predadores sexuais. Como exemplo, pode-se citar algumas palavras que denotam raiva (perturbador e furioso), tristeza (chateado e triste), alegria (feliz e alegre) e medo (assustado e pânico).

Ressalta-se que não somente a presença dessas palavras é suficiente para a detecção, mas outros aspectos também presentes nos textos são aplicados, como, por exemplo, a recusa em mudar de assunto durante um chat que os predadores sexuais apresentam. Tal característica é chamada em Bogdanova (2012) de *Fixated Discourse*.

Outra característica notável nos chats estudados do PJ é a introdução, em determinado momento da conversa, de palavras de dessensibilização, ou seja, palavras que expressam uma mudança radical de intenção do então colega de bate-papo. Assim, se a intenção final é conseguir atos de sexo com menores, em certo momento e sem sutileza, palavras que denotam realmente a intenção do predador são colocadas na conversa.

2.7. Ferramentas Utilizadas

Para a realização deste trabalho de graduação as seguintes ferramentas foram utilizadas:

2.7.1- Editor Notepad++ v6.8.6

O editor Notepad++ é um poderoso editor de textos disponível gratuitamente para download e foi aplicado nas etapas de limpeza e pré-processamento onde os scripts em linguagem Python não foram suficientes para eliminar de modo automatizado todo o conteúdo não útil presente nos *logs* coletados do PJ e do CyberSex.

2.7.2- Python IDLE 2.7.10

O IDLE é um ambiente simples para programação em linguagem Python e foi aplicado neste trabalho nas etapas de raspagem de dados ou *data scraping*. Foi usado devido à gratuidade da licença, por ser simples o uso e por possuir vasta gama de bibliotecas como, por exemplo, as bibliotecas BeautifulSoup, Urllib2, NLTK (Natural Language ToolKit) e Re que foram aplicadas no download, extração e tratamento das mensagens raspadas do PJ.

2.7.3- SentiWordNet 3.0

O SentiWordNet é um recurso léxico usado para mineração de opinião e tem por base o WordNet– vasta base de dados de substantivos, verbos, adjetivos e advérbios em Inglês que são agrupados em conjuntos de sinônimos cognitivos (*synsets*) -, e é usado para atribuir para cada *synset* três valores de sentimentos: valores positivos, valores negativos e valores neutros, denotando respectivamente a positividade, a negatividade ou a objetividade de cada *synset*.

O referido recurso foi aplicado neste trabalho de graduação coma finalidade de extrair os valores de positividade, negatividade ou objetividade dos textos monitorados.

2.7.4- WordNet-Affect – Strapparava (2004)

O WordNet-Affect é uma extensão léxica usada para representação de conhecimento afetivo. A partir do WordNet³, o WordNet-Affect foi desenvolvido por meio da seleção e rotulação dos *synsets* representando assim os conceitos afetivos.

³ O WordNet é uma vasta base de dados léxica de substantivos, verbos, adjetivos e advérbios em Inglês que são agrupados em conjuntos de sinônimos (*synsets*), sendo que cada conjunto expressa um conceito particular.

Esta solução foi utilizada para a classificação e pontuação das palavras cujos conceitos afetivos são marcantes na personalidade dos predadores.

2.7.5- IDE Eclipse Java Versão Luna 2(4.4.2)

O Eclipse, muito além de ferramenta de programação, apresenta-se como um ambiente completo de desenvolvimento utilizando a linguagem Java. A linguagem Java foi utilizada devido à facilidade na programação e também pela existência de uma gama muito vasta de *frameworks*, *plug-ins* e bibliotecas.

2.7.6- Apache Maven 3.3.3

O Maven é, essencialmente, um gerenciador de *builds* e dependências baseado na arquitetura POM (*Project Object Model*). As dependências de algumas bibliotecas do presente trabalho foram resolvidas com o auxílio desta ferramenta.

2.7.7- Biblioteca JavaMail 1.5.3

O JavaMail é uma biblioteca utilizada para envio e recebimento e-mails. A referida biblioteca foi aplicada no corrente trabalho devido à facilidade na implementação e também pelo motivo de existir muita informação e exemplos de uso.

2.7.8- Biblioteca EclipseLink 2.5.2

O EclipseLink é uma biblioteca usada para persistência e mapeamento Object-Relational-Mapping (ORM). O uso da referida biblioteca permite um grau elevado de abstração quando da aplicação de métodos para manipulação de bases de dados dentro de um ambiente orientado a objetos, e seu uso deve-se ao fato da facilidade de implementação e a vasta existência de material de apoio.

2.7.9- JavaFX 2.0

O JavaFX é um conjunto de pacotes de mídia e gráficos que permite aos desenvolvedores criar, testar, depurar e distribuir aplicações para diversas plataformas. Foi usado neste trabalho de graduação devido à facilidade de implementação e na criação da interface visual do sistema.

2.7.10- Scene Builder 2.0

O Scene Builder é uma ferramenta para criação de *layouts* visuais para aplicações JavaFX. Foi aplicado neste trabalho de graduação na criação da interface visual do sistema e devido ao tipo de licença e facilidade de uso.

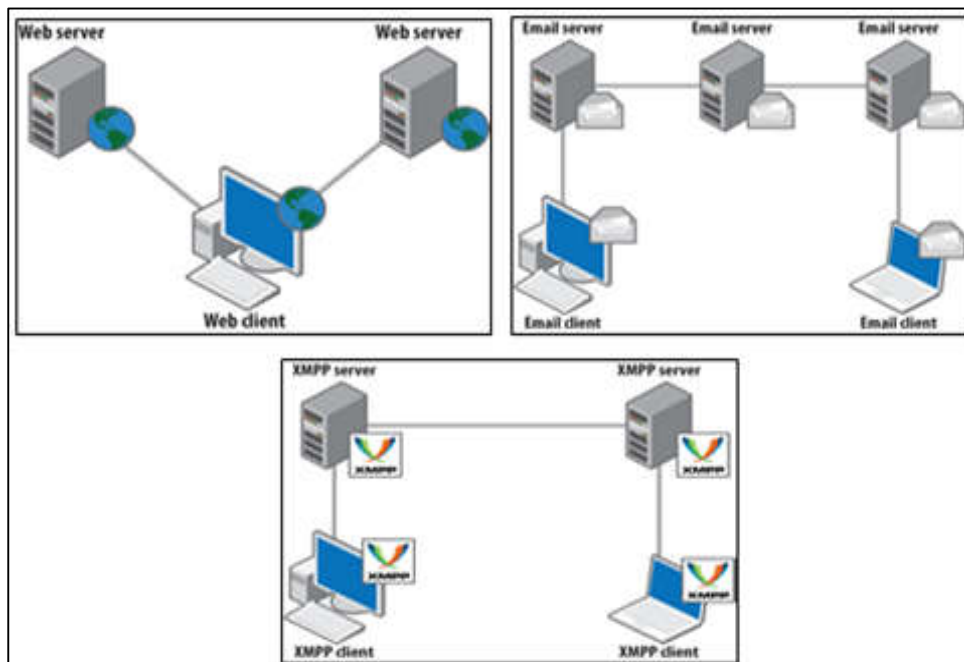
2.7.11- Biblioteca Smack versão 4.1.1

A biblioteca Smack permite a decodificação das mensagens do tipo XMPP (eXtensible Messaging and Presence Protocol) oriundas dos servidores do Facebook. Em essência, essa biblioteca faz *streaming* de arquivos XML (eXtensible Markup Language), permitindo o envio e recebimento de mensagens e de informações de *status* do usuário.

A arquitetura dos sistemas que empregam o XMPP é do tipo cliente-servidor similar às arquiteturas de serviços web e e-mail. A diferença fundamental reside no fato de que as conexões entre servidores XMPP são diretas (*single hop*) devido à necessidade de rapidez na comunicação.

Na Figura 11, são apresentadas as arquiteturas cliente-servidor web, e-mail e XMPP.

Figura 11 - Arquiteturas básicas cliente-servidor web, e-mail e XMPP



Fonte: Saint-Andre (2009)

A diferença arquitetural básica que, dentre outros fatores, garante comunicações realmente velozes está na comunicação entre servidores, pois, mesmo quando os clientes XMPP estão em domínios diferentes, não há servidores intermediários.

2.7.12- MySQL Server 5.6 e MySQL Workbench 5.6 CE

O MySQL Server 5.6 é um banco de dados padrão ANSI SQL com funcionalidades extras e o Workbench 5.6 CE é um IDE para gerenciamento de bases de dados MySQL. O referido SGBD foi escolhido devido à facilidade de implementação e também pelo tipo de licença.

2.7.13- OpenNLP 1.6.0

A OpenNLP é uma biblioteca da Apache utilizada para processamento de textos em linguagem natural para aplicações de linguagem de máquina. Foi aplicada neste trabalho de graduação devido à facilidade de implementação e pelo tipo de licença.

2.8. Soluções e Pesquisas Existentes

Não existem comercialmente mecanismos de detecção capazes de extrair de sites de bate-papo conhecimento relevante e de inferir comportamentos em tempo real e de forma automatizada. No entanto, existem alguns programas e scripts que executam parte das tarefas propostas neste trabalho de graduação.

De acordo com Morris (2013) em seu artigo *Identifying Online Sexual Predators by SVM Classification with Lexical and Behavioral Features* tem-se a proposta de um método para descobrir predadores sexuais em uma “coleção” de chats. Neste caso, a automatização é mais abrangente, pois não se limita a um site específico.

Essencialmente, o método de Morris (2013) aplica o conceito de *Support Vector Machines* (SVM) - técnica de aprendizado de máquina (AM) de uso crescente na comunidade científica (LORENA, 2007) - usando contagens do tipo *n-gram*⁴ com *n* igual a um (*n*=1) e também com *n* igual a dois (*n*=2) tanto para os datagramas gerados com os dados do agressor como aqueles gerados com dados da vítima.

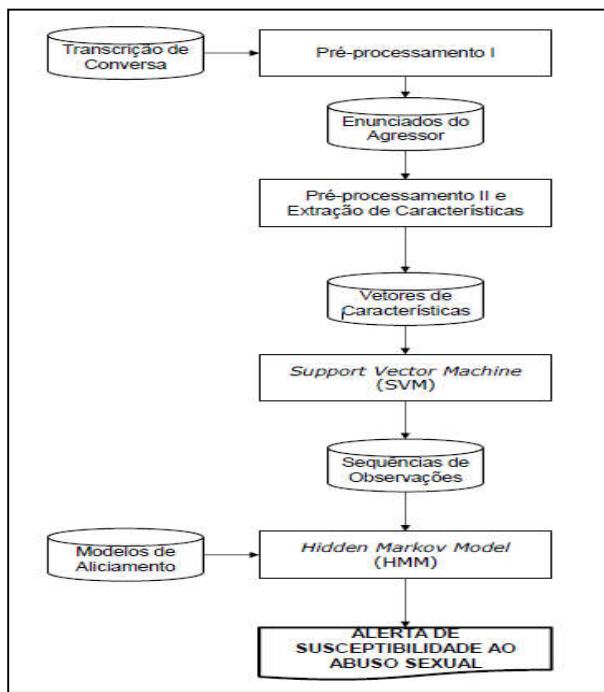
⁴ *n-gram*: subsequência de quaisquer *n* elementos adjacentes em uma sequência de símbolos (letras, números, palavras ou sílabas).

O referido método foi criado e submetido ao “*sexual predator task*” no laboratório da PAN 2012 “*Uncovering Plagiarism, Authorship, and Social Software Misuse*”, mostrando que a falta de sistemas prontos e eficazes para o reconhecimento automático de padrões impróprios de conversação tem lançado para a comunidade científica algumas competições que são disputadas com a finalidade de fomentar a pesquisa acerca de problemas relacionados com detecção de plágio, autoria e, no contexto deste trabalho, o mau uso de *software* social.

Conforme Pendar (2007) e McGhee (2011) a aplicação do conceito de SVM aliado ao algoritmo de classificação K-NN cuja finalidade é usar uma base nas quais os dados são separados em várias classes para prever a classificação de um novo dado amostrado, mostrou-se razoavelmente bem atingindo cerca de 80% de acertos tendo como base, também, os *logs* do PJ.

De acordo com Santin (2013), apesar de ter usado técnica semelhante às encontradas na literatura, a aplicação de classificação baseada no conceito de SVM usando somente os dados do agressor e aplicando o modelo HMM (Hidden Markov Model) mostrou-se mais eficaz tendo em vista a necessidade de rapidez na detecção de padrões. Na Figura 12 tem-se o sequenciamento do processo de detecção proposto por Santin (2013).

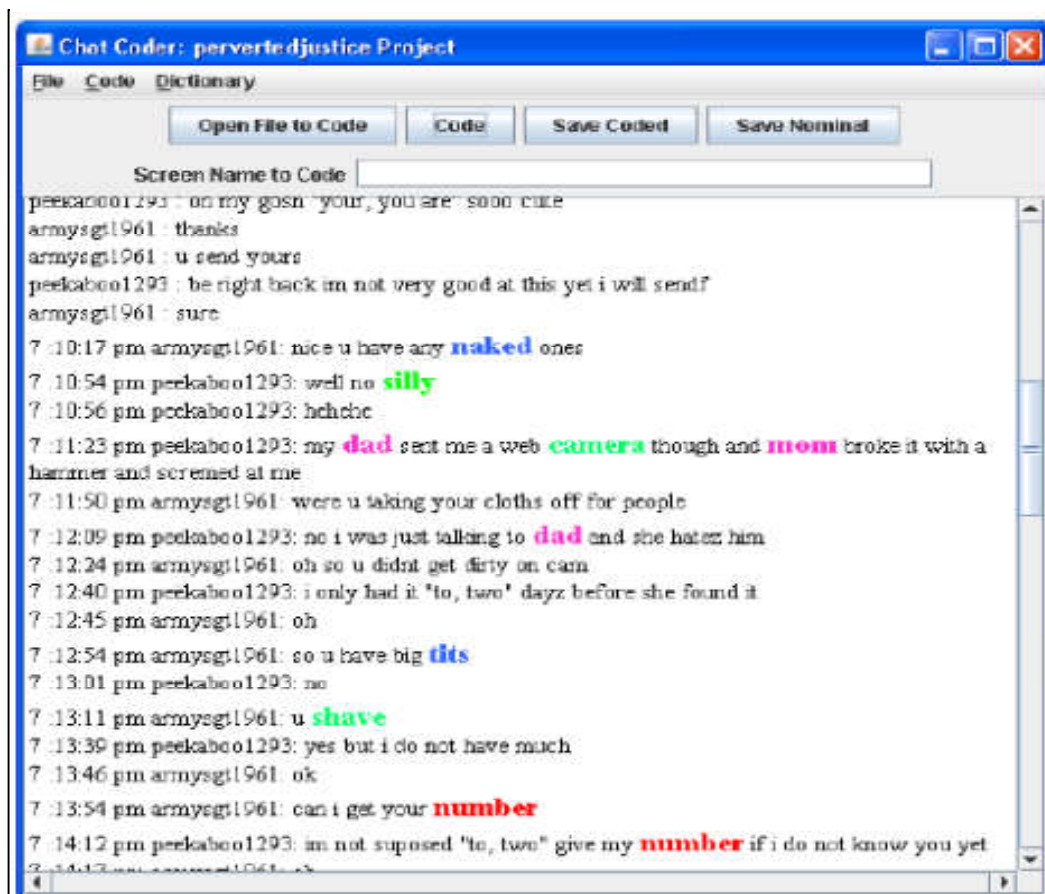
Figura 12 – Sequenciamento - somente agressor com modelo HMM



Fonte: Santin (2013)

Apesar de não haver ainda um software pronto para uso, Kontostathis (2008) apresenta um trabalho sobre um programa chamado Chatcoder. A Figura 13 apresenta uma das telas do referido programa com um *log* transcrito do PJ, podendo-se notar uma codificação por cores no que se refere às classes dos vocábulos.

Figura 13 – ChatCoder com Logs do PJ

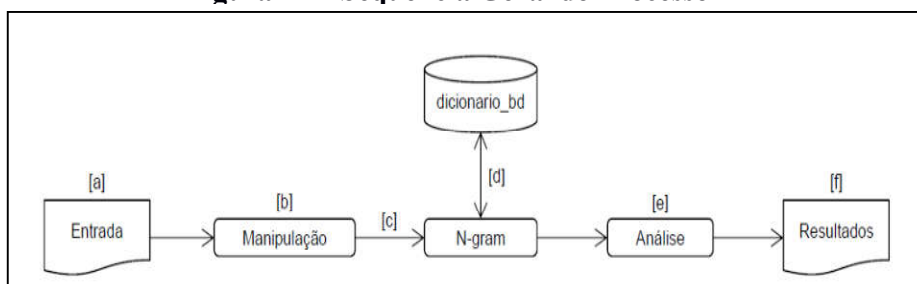


Fonte: Kontostathis (2010)

O referido programa é essencialmente uma ferramenta de mineração de textos⁵ para análise de *logs* transcritos de chats (*logs* tratados). Existe também a possibilidade de ser usado de forma automatizada para extrair códigos de todos os arquivos de *log* de um diretório – *batch mode*.

Na Figura 14 sugere-se uma sequência lógica do mecanismo completo de captura, extração, análise e apresentação de resultados de acordo com Santin (2013).

⁵ Processo de Descoberta de Conhecimento que utiliza técnicas de análise e extração de dados a partir de textos, frases ou apenas palavras.

Figura 14 – Sequência Geral do Processo

Fonte: Santin (2013)

A sequência mostrada na Figura 3 representa a entrada do texto conforme captura em [a]; a manipulação do texto em [b]; o envio dos *datagramas* extraídos para confronto/verificação com a base de palavras em [d]; a análise do confronto em [e]; e a apresentação dos resultados em [f].

De acordo com Bogdanova (2012) a análise de sentimentos com foco não somente na distinção entre vítima e predador, mas também nas instabilidades emocionais reveladas pelos pedófilos revelou-se de grande valia atingindo acertos na ordem de 90%.

Tem-se no referido trabalho um direcionamento distinto e extra que é a análise de sentimentos com base na classificação de palavras como negativas positivas ou objetivas tendo com base forte as características emocionais dos pedófilos.

Na Tabela 1 constam alguns materiais acadêmicos - pesquisas e *softwares* encontrados acerca do tema proposto neste trabalho de graduação. No cabeçalho das colunas constam características e funcionalidades presentes no corrente trabalho de graduação.

Tabela 1- Soluções e Pesquisas Existentes

TÍTULO	MONITORA ASSÉDIO SEXUAL	MONITORAÇÃO EM TEMPO REAL	ANÁLISE DE SENTIMENTOS	SOFTWARE	ALARME REMOTO
CHATCODER	X	X	X	✓	X
A Two-step Approach for Effective Detection of Misbehaving Users in Chats	✓	X	✓	X	X
Análise Automática de Textos de Mensagens Instantâneas para Detecção de Aliciamento Sexual de Crianças e Adolescentes	✓	X	✓	X	X
BULLYTRACKER Detecting the Presence of Cyberbullying Using Computer <i>Software</i>	X	X	X	✓	X

Fonte: o autor

3- DESENVOLVIMENTO

Este capítulo apresenta o desenvolvimento da solução proposta na forma de duas arquiteturas distintas: uma arquitetura aplicada para treinamento, validação e checagem da RNA, a qual envolve as tarefas de KDD aplicadas aos dados que, no contexto deste trabalho de graduação, são os *logs* de predadores e não predadores, e uma arquitetura aplicada para a detecção dos predadores sexuais com a aplicação da RNA previamente treinada.

3.1- Arquitetura de Treinamento

A arquitetura de treinamento aplicada para a preparação da RNA é explicada seguindo-se as fases presentes no processo de KDD.

3.1.1- Seleção (*Download*)

A presente etapa de KDD restringiu-se a escolha de 3 fontes de *logs*: o PJ, a NPS (*Naval Postgraduate School*) e o Cybersex Log Archive. Dos sites do PJ e do CyberSex, foram coletados todos os 606 *logs* disponíveis (593 do PJ e 13 do CyberSex) por meio da aplicação de um script criado em linguagem Python usando uma técnica conhecida como *data scraping* (raspagem de dados).

Na Figura 15 segue um trecho do script em linguagem Python denominado aqui script1 onde são apresentadas as bibliotecas utilizadas. A biblioteca `urllib2` foi responsável por fornecer os métodos necessários para o download do conteúdo bruto das páginas de *logs*.

Figura 15- Bibliotecas do Script1

```
# Bibliotecas utilizadas #

from __future__ import print_function
import MySQLdb
from bs4 import BeautifulSoup as BS
# bibliotecas não nativas:
# urllib2, BeautifulSoup e mysql
from datetime import date, datetime, timedelta
import re, urllib2, time
```

Fonte: o autor

Na Figura 16 segue outro trecho do script1 apresentando o código usado para o download da lista de arquivos dos predadores condenados com a ajuda do PJ. Nota-se que já há uma prévia limpeza nos dados com o uso do método `findAll` da biblioteca BeautifulSoup.

Figura 16- Trecho de Download dos Logs do PJ do Script1

```
htmlNames = urllib2.urlopen("http://www.perverted-justice.com/?con=full")
# Nesta pagina encontra-se a lista
# como todos os predadores condenados
# com ajuda do PJ - 592 total
soupNames = BS(htmlNames.read())

for ahref in soupNames.findAll('a', attrs={'id': 'pedoLink'}):
    for nome in ahref:
        nomes.append(nome)
    listaNomes = list(set(nomes))
    tamanho = len(listaNomes)
    for nome in listaNomes:
        argNomes.write(nome+"\n")
```

Fonte: o autor

Os logs do CyberSex foram salvos aplicando o script1 com alterações no URL (Uniform Resource Locator) passado como argumento no método da biblioteca `urllib2`, porém não foi criada nenhuma lista de nomes e, assim, o argumento foi alterado para cada um dos 13 logs disponíveis. Alguns destes 13 logs do CyberSex eram oriundos de chats com mais de 2 participantes e, por essa razão, estes foram manualmente editados totalizando finalmente 28 arquivos diferentes.

Com relação aos logs da NPS, estes foram selecionados por meio de outro script denominado aqui script2 também em linguagem Python com o uso de uma biblioteca chamada NLTK. Para estes logs não foi necessário efetuar o download, pois este *corpus*⁶ encontra-se incorporado à biblioteca. Por meio de alguns métodos importados da referida biblioteca, os logs foram salvos para posterior aplicação.

3.1.2- Extração / Limpeza

De modo a entregar todos os logs na etapa seguinte (transformação), foram aplicados alguns métodos do script1 após o download dos logs do PJ e do CyberSex. Nesta fase de KDD, a matéria bruta foi manipulada e limpa para que se pudesse extrair os dados necessários dentre todo o conjunto formado pelo código-fonte da

⁶ *Corpus* significa coletânea ou conjunto de documentos sobre determinado tema. Na biblioteca NLTK se usa esta terminologia quando da aplicação destes conjuntos.

página e pelo conteúdo dos chats. A Figura 17 apresenta um trecho do script1 contendo o código responsável por parte da limpeza dos *logs*.

Figura 17- Limpeza dos *Logs*

```
str = listaNomes[i]+"&noComm=true"
url = "http://www.perverted-justice.com/index.php?archive="+str
htmlLogs = urllib2.urlopen(url)
soupLogs = BS(htmlLogs.read())
for code in soupLogs.find_all('span', attrs={'class': 'code_c'}):
    code.replace_with("")
for chat in soupLogs.find_all('div', attrs={'class': 'chatLog'}):
    chat.contents[0].replace_with("")
```

Fonte: o autor

O método `find_all` da biblioteca BeautifulSoup permitiu, com certo grau de abstração, remover grande parte das *tags* do código-fonte e assim limpar os dados indesejados.

Com relação aos *logs* do PJ, dos 593 *logs* disponíveis, somente 518 foram totalmente tratados e disponibilizados como arquivos úteis. Sobre os *logs* da NPS a limpeza foi mais branda, pois foi necessária a remoção de somente três palavras identificadoras (ACTION, JOIN e PART) constantes das saídas produzidas pelo método de extração aplicado no script2. Na Figura 18 tem-se um trecho do código responsável pela limpeza dos *logs* da NPS.

Figura 18 - Limpeza nos *Logs* da NPS

```
for post in nps.posts(fid):
    line = ' '.join(post).rstrip()
    if 'ACTION' in line or 'JOIN' in line or 'PART' in line:
        continue
```

Fonte: o autor

3.1.3- Transformação

Nesta etapa, cada *log* foi armazenado em disco rígido na forma de um arquivo único, formando um conjunto de 561 arquivos (518 do tipo PJ predador, 28 do tipo CyberSex não predador e 15 do tipo NPS não predador).

A transformação aplicada nesta fase começou com o acesso aos arquivos limpos e, para cada arquivo dos três tipos, foram utilizados métodos em linguagem

Java da biblioteca OpenNLP (*Opensource Natural Language Processing*) que aplicam uma *tag* (etiqueta) a cada sequência de caracteres constantes do arquivo e de acordo com sua categoria gramatical na Língua Inglesa.

Para que se conseguisse essa espécie de classificação foi necessário, antes da aplicação dos métodos, que se usasse um modelo de dados para referência e o modelo usado nesta etapa foi o “*en-pos-maxent.bin*”, um modelo de classificação na Língua Inglesa (*en*), do tipo *Part-Of-Speech tagging* - partes do discurso ou partes da oração (*pos*) - e que usa o método de máxima entropia⁷ (*maxent*).

Como exemplo, o termo *car* recebe como *tag* a sequência *_NN*, sendo assim alterado para *car_NN*. A *tag* *_NN* identifica um termo como substantivo (*noun* em Inglês). Na Figura 19 é apresentado um método chamado *tagging* criado com a finalidade de transformar cada palavra dos *logs* em palavras adequadas aos métodos das fases posteriores.

O retorno do método *tagging* é um novo arquivo com todas as palavras etiquetadas conforme suas classificações na gramática da Língua Inglesa.

Figura 19 –Classificação - POStagging

```
public class PostTagger {
    // Aplicação de tags - POSTag classification
    public File tagging(File file, String pathToSave) throws IOException {
        File fileOut = new File(pathToSave + "FilePOSTag_" + file.getName());
        POSModel model = new POSModelLoader().load(new File("Models/en-pos-maxent.bin"));
        POSTaggerME tagger = new POSTaggerME(model);
        BufferedReader br = new BufferedReader(new FileReader(file));
        BufferedWriter bw = new BufferedWriter(new FileWriter(fileOut));
        String line, tags[];
        while ((line = br.readLine()) != null) {
            String whitespaceTokenizerLine[] = WhitespaceTokenizer.INSTANCE.tokenize(line);
            tags = tagger.tag(whitespaceTokenizerLine);
            POSSample sample = new POSSample(whitespaceTokenizerLine, tags);
            bw.write(sample.toString().toLowerCase() + "\n");
        }
        bw.close();
        return fileOut;
    }
}
```

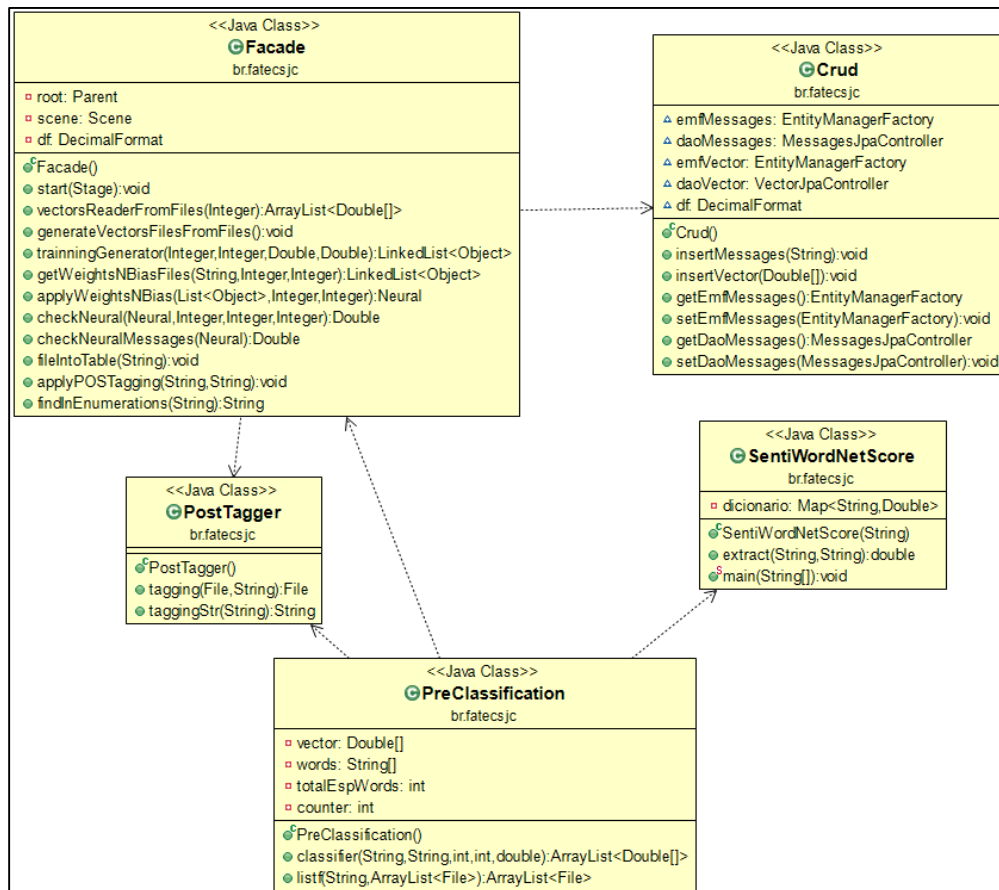
Fonte: o autor

A fim de preparar os dados para a próxima fase, cada arquivo com conteúdo classificado ou etiquetado foi transformado em um vetor do tipo *Double* com tamanho 18. A primeira posição foi usada para guardar um valor que pode ser positivo, negativo ou neutro conforme o resultado do método de cálculo de sentimentos aplicado ao arquivo completo e pode variar de -1,0 até+1,0.

⁷ Segundo Cassetari (2003) a Máxima Entropia é uma lei que reflete a tendência natural das coisas de se aproximar do estado caótico a menos que algo atue sobre elas para evitar isso.

A Figura 20 apresenta um diagrama contendo as classes envolvidas nesta fase de transformação de dados. Um padrão Facade simples foi criado a fim de se ter em um só lugar os métodos usados constantemente durante esta etapa.

Figura 20 - Diagrama de Classes – Etapa de Transformação



Fonte: o autor

Para o cálculo do valor da primeira posição do vetor, cada palavra dos arquivos não classificados foi lida e comparada com o conteúdo do SentiWordNet usando métodos POJO⁸ criados para calcular e acumular o valor de sentimento de cada palavra.

Com o total acumulado de cada arquivo, procedeu-se ao cálculo da média aritmética dividindo-se o total acumulado pelo total de palavras lidas. Da 2ª até a 17ª posição do vetor, foram guardados valores referentes à porcentagem da quantidade de palavras encontradas em cada arquivo com relação ao total de palavras dos tipos descritos no trabalho de Bogdanova (2012).

⁸ Sigla da frase: *Plain Old Java Object* significa que o código referencia objetos que não dependem de classes ou bibliotecas externas.

Na 18ª posição, foi guardado o tipo de arquivo no qual foi aplicada a transformação, sendo o valor 1,0 para predador e 0,0 para não predador. Finalmente, tem-se como resultado desta etapa de transformação do processo de KDD um vetor do tipo Double, com tamanho 18 para cada arquivo de *log*. Na Tabela 2 encontra-se a descrição completa do vetor.

Tabela 2- Descrição do Vetor de Entrada da RNA

Posição	Descrição
1	Positividade/Objetividade/Negatividade do texto [-1,0;1,0]
2	Palavras que expressam alegria (%)
3	Palavras que expressam tristeza (%)
4	Palavras que expressam raiva (%)
5	Palavras que expressam surpresa (%)
6	Palavras que expressam desgosto (%)
7	Palavras que expressam medo (%)
8	Palavras que expressam aproximação (%)
9	Substantivos que expressam relacionamento (%)
10	Palavras que expressam família (%)
11	Palavras que expressam dessensibilização (%)
12	Palavras que expressam informação (%)
13	Pronomes pessoais (%)
14	Pronomes reflexivos (%)
15	Verbos que expressam obrigação (%)
16	Emoticons (%)
17	Sentenças imperativas (%)
18	Categoria (1.0 - predador 0.0 - não predador)

Fonte: o autor

Para fins de salvaguarda de dados, cada vetor gerado foi inserido em uma tabela de uma base de dados do tipo MySQL chamada fbchatwordlog. Os vetores gerados como resultado da transformação dos *logs* limpos do PJ foram guardados na tabela vectorpred. Os vetores resultantes dos *logs* da NPS foram inseridos na tabela vectornps e, por fim, os vetores resultantes dos *logs* do CyberSex foram salvos na tabela vectorcyber.

Para a persistência dos vetores no banco de dados foram aplicados métodos JPA da biblioteca EclipseLink juntamente com alguns outros métodos POJO. Na Tabela 3 encontra-se o dicionário de dados das referidas tabelas.

Tabela 3- Dicionário de Dados das Tabelas de Vetores

Atributo	Tipo de dado	Descrição
id	Inteiro (11 dígitos)	Chave de identificação do vetor
vector	Texto(1000 caracteres)	Vetor na forma de uma linha de texto

Fonte: o autor

3.1.4- Mineração dos Dados – Busca por Padrões

Para esta etapa de KDD, foram selecionados vários vetores para aplicação na RNA com a finalidade de treinamento e validação. A fim de se proporcionar certo grau de balanceamento, de todos os vetores disponíveis, 80 foram selecionados, sendo 15 do tipo NPS, 40 do tipo PJ e 25 do tipo CyberSex.

Foram selecionados 20 vetores do tipo PJ, 10 do tipo CyberSex e 10 do tipo NPS para treinamento. Para validação foram selecionados 10 vetores do tipo PJ, 5 do tipo CyberSex e 5 do tipo NPS. Com essa divisão formou-se um conjunto de treinamento com 40 e um conjunto de validação com 20 vetores.

Para o conjunto de testes foram selecionados 10 vetores do tipo PJ, 10 do tipo CyberSex e os mesmos 5 vetores do tipo NPS utilizados para validação. Assim, a estratégia de treinamento, validação e teste foi de aproximadamente 50% para treinamento, 28% para validação e 22% para teste.

Para que se procedesse ao treinamento e validação a RNA foi previamente configurada com os seguintes valores:

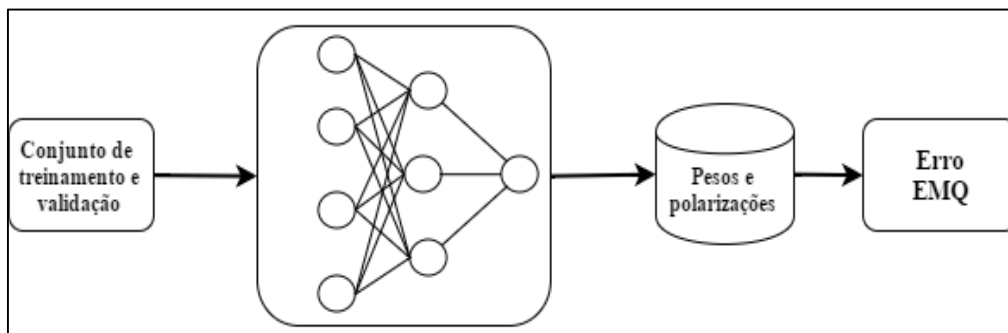
- Quantidade de entradas: 17;
- Quantidade de neurônios da camada oculta: 5;
- Taxa de aprendizado: 0,1;
- Momento: 0,1;
- Quantidade de épocas: 10000;
- Saídas: 1; e
- Arquitetura: MLP.

A quantidade de entradas da rede é informada conforme o tamanho total do vetor mesmo tendo sido guardada a categoria na última posição, pois a RNA é configurada para reconhecer a última posição como saída a alcançar.

Na Figura 21 é apresentada a sequência geral utilizada para treinamento e validação da rede, contendo um conjunto de treinamento e validação, a RNA, o

backup dos pesos e polarizações em disco rígido e, finalmente, o cálculo do erro médio quadrático (EMQ).

Figura 21 – Sequência de Treinamento e Validação



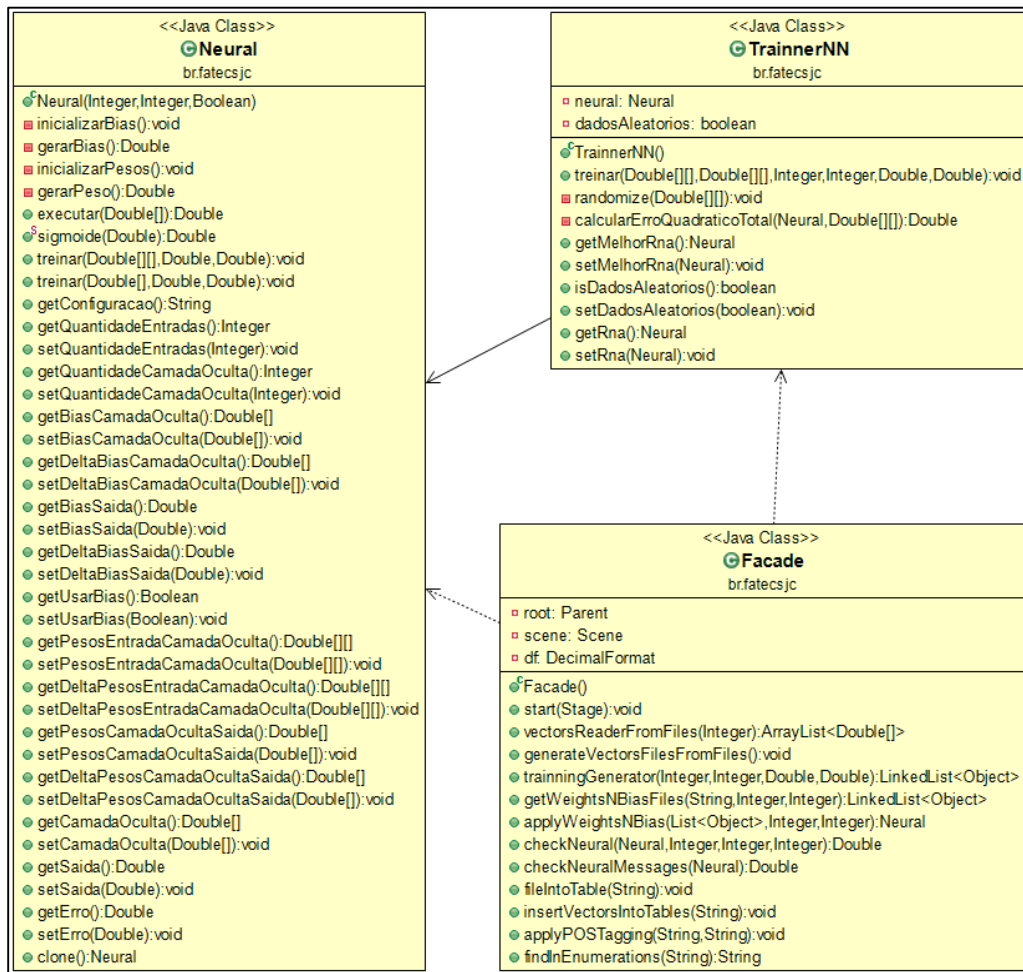
Fonte: o autor

Na Figura 22 tem-se um diagrama contendo as classes *Trainer* e *Neural* responsáveis pela implementação da RNA e a classe *Facade* que contém, além de outros, os métodos utilizados para treinamento e validação. A estratégia utilizada para a divisão dos *logs* foi a seguinte: treinamento com 56% e validação com 28%.

O método *trainingGenerator* da classe *Facade* aplica os conjuntos de treinamento e validação preestabelecidos, a quantidade de épocas, a taxa de aprendizado e o fator de momentum ao método *treinar* da classe *Trainer*.

Após cada treinamento, todos os pesos e polarizações (*bias*) gerados durante a execução foram salvos em arquivos no disco rígido, permitindo-se assim recuperar as informações de configuração da RNA e usá-las como um padrão a seguir.

Figura 22 – Treinamento / Testes



Fonte: o autor

Na Figura 23 é apresentado o código usado na classe MainApp para treinar e validar a RNA. Para cada vetor lido e aplicado à RNA, o erro foi calculado e acumulado na variável *acc* para que ao final o erro médio quadrático fosse calculado dividindo-se o valor acumulado pelo valor da variável *cont*.

Cada EMQ calculado foi guardado para comparações com os erros resultantes da aplicação do mesmo processo, porém com alterações nos argumentos passados aos métodos da rede neural.

Figura 23 – Treinamento e Validação

```

public static void main(String[] args) throws Exception {
    DecimalFormat df = new DecimalFormat("0.00000");
    Toolkit.getDefaultToolkit().beep();
    Facade facade = new Facade();
    Integer esperado = 1, entradas = 17, co = 5, epocas = 10000, cont = 0, iter = 0;
    Double ta = 0.1, fm = 0.1;
    Double saida = 0d, saidaArbitrada=0.17442, erroAbsoluto, erro, limiarAtribuido = 0.1;
    Double acc = 0d, pot = 0d, emq = 1d;
    System.out.println("Teste RNA\nLimiar atribuido="+limiarAtribuido+"\nNeurônios CO="+co+
        "\nEpocas="+epocas+"\nTaxa Aprendizado="+ta+"\nFator de momentum="+fm);
    while (emq > saidaArbitrada){
        ArrayList<Object> weightsBiasTraining = facade.trainningGenerator(co, epocas, ta, fm);
        neural = facade.applyWeightsNBias(weightsBiasTraining, entradas,co);
        for (int i = 0; i < 31; i++){
            cont++;
            saida = facade.checkNeural(neural,i,i,4);
            erro = esperado-saida;
            acc += Math.pow(erro,2);
        }
        emq = acc / cont;
        System.out.println("EMQ - "+df.format(emq));
    }
    Toolkit.getDefaultToolkit().beep();
    System.exit(0);
}

```

Fonte: o autor

Tem-se na Tabela 4 vários treinamentos efetuados com o conjunto de treinamento e validação completo (31 vetores). A configuração inicial da rede foi mantida alterando-se somente o fator de momentum. Outros valores de quantidade de neurônios na camada oculta e taxa de aprendizado foram testados, porém os erros foram elevados. Portanto, na Tabela 4 encontram-se as melhores combinações de ajustes da RNA.

Tabela 4 – Ajustes da RNA para Treinamento e Validação

Ajuste	Vetores	Neurônios na camada oculta	Taxa de aprendizado	Fator de momentum	Épocas	EMQ
1	31	5	0,1	0,1	10000	0,17442
2	31	5	0,1	0,2	10000	0,18591
3	31	5	0,1	0,3	10000	0,18146
4	31	5	0,1	0,4	10000	0,17968
5	31	5	0,1	0,5	10000	0,17906
6	31	5	0,1	0,6	10000	0,17833
7	31	5	0,1	0,63	10000	0,17836
8	31	5	0,1	0,7	10000	0,17835
9	31	5	0,1	0,8	10000	0,17822
10	31	5	0,1	0,9	10000	0,17819
11	31	5	0,1	0,95	10000	0,17817

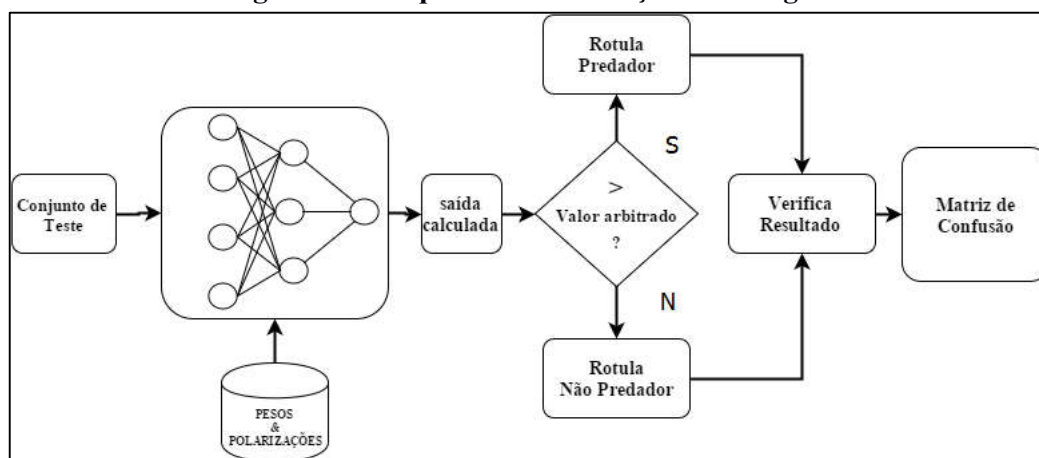
Fonte: o autor

Nota-se na Tabela 4 que o melhor resultado obtido encontra-se na coluna Ajuste, linha 1 com EMQ de 0,17442.

3.1.5- Avaliação / Checagem

Com o menor EMQ encontrado e tendo guardada a configuração da RNA, procedeu-se com a checagem e avaliação dos resultados. Na Figura 24 encontra-se o procedimento utilizado para checagem e avaliação dos resultados.

Figura 24 – Sequência de Avaliação / Checagem



Fonte: o autor

O processo aplicado para a checagem descrito na Figura 24 resume-se em aplicar à RNA o conjunto de teste – vetor por vetor – e verificar a saída apresentada para cada vetor. Se o valor de saída calculado pela RNA for maior que um valor arbitrado, tem-se que o vetor aplicado é do tipo predador. Se o valor de saída calculado for menor ou igual ao valor arbitrado, o vetor aplicado é classificado como do tipo não predador. Após essa classificação imposta, os resultados são colocados na matriz de erro para verificação da taxa de acertos.

Como exemplo, se um vetor do tipo não predador gera na saída da rede um valor menor do que o arbitrado, então se considera que o tipo do vetor é não predador e, nesse caso, a rede neural acertou. Na Figura 25 encontram-se as matrizes de erro ou confusão com os resultados da checagem para diferentes valores arbitrados.

Figura 25 – Matrizes de Confusão

0,1	Pred	Não Pred	Total linha	0,2	Pred	Não Pred	Total linha	0,3	Pred	Não Pred	Total linha
Pred	12	1	13	Pred	11	2	13	Pred	10	3	13
Não Pred	11	7	18	Não Pred	6	12	18	Não Pred	4	14	18
Total coluna	23	8	31	Total coluna	17	14	31	Total coluna	14	17	31
			61,29%				74,19%				77,42%
0,4	Pred	Não Pred	Total linha	0,5	Pred	Não Pred	Total linha	0,6	Pred	Não Pred	Total linha
Pred	10	3	13	Pred	8	5	13	Pred	7	6	13
Não Pred	3	15	18	Não Pred	3	15	18	Não Pred	1	17	18
Total coluna	13	18	31	Total coluna	11	20	31	Total coluna	8	23	31
			80,65%				74,19%				77,42%
0,7	Pred	Não Pred	Total linha	0,8	Pred	Não Pred	Total linha	0,9	Pred	Não Pred	Total linha
Pred	7	6	13	Pred	5	8	13	Pred	3	10	13
Não Pred	1	17	18	Não Pred	0	18	18	Não Pred	0	18	18
Total coluna	8	23	31	Total coluna	5	26	31	Total coluna	3	28	31
			77,42%				74,19%				67,74%

Fonte: o autor

Observou-se que, com um valor arbitrado de 0,4 para o limiar de classificação, a relação de acertos/total chegou a 80.65%, ou seja, do total de 31 vetores aplicados no teste 25 foram corretamente classificados. Segue na Figura 26 o código utilizado para verificar o erro resultante de cada vetor do conjunto de teste.

Figura 26 – Código de Avaliação / Checagem

```

public static void main(String[] args) throws Exception {
    DecimalFormat df = new DecimalFormat("0.00000");
    Toolkit.getDefaultToolkit().beep();
    Facade facade = new Facade();
    Integer esperado = 1, entradas = 17, co = 5, epocas = 10000, cont = 0, iter = 0;
    Double ta = 0.1, fm = 0.1;
    Double saida = 0d, saidaArbitrada=0.5;
    System.out.println("Teste RNA\nLimiar atribuido="+saidaArbitrada);
    ArrayList<Object> weightsBiasChecking = facade.getWeightsNBiasFiles("Output", entradas, co);
    neural = facade.applyWeightsNBias(weightsBiasChecking, entradas,co);
    System.out.println("VETOR\tESPERADO\tSAÍDA\t\tCLASSIFICADO\tREAL\t\tRESULTADO");
    for (int i = 0; i < 31; i++){
        cont++;
        esperado = i<18?0:1;
        saida = facade.checkNeural(neural,i,i,4);
        String real = null, esp = null;
        System.out.println(
            cont+"\t"+esperado+"    "+" \t\t"+df.format(saida)+
            "\t\t"+(esp=saida>saidaArbitrada?"Pred":"NãoPred")+
            "\t\t"+(i<18?real="NãoPred":(real="Pred"))+
            (esp.equals(real)?"\t\tAcertou":"\t\tErrou"));
    }
    Toolkit.getDefaultToolkit().beep(); System.exit(0);
}

```

Fonte: o autor

3.2- Arquitetura de Detecção

A arquitetura de detecção do Chatlisten é apresentada primeiramente na forma de um quadro sinótico na Figura 27. Começando na parte superior, tem-se uma nuvem representando conjuntamente a Internet e os servidores de chat do Facebook. Nota-se que o sistema propõe uma monitoração passiva, pois somente recebe as mensagens oriundas do usuário monitorado com destino ao usuário alvo.

Na sequência, o usuário do sistema - pessoa que conhece as credenciais de acesso do usuário alvo (criança ou adolescente que se quer monitorar) - pode ativar ou não o recebimento das mensagens que dão origem às demais etapas do processo de detecção. Para que se dê início ao recebimento das mensagens, o usuário do sistema deve inserir em campos apropriados na tela do sistema o *login* e a senha do usuário alvo.

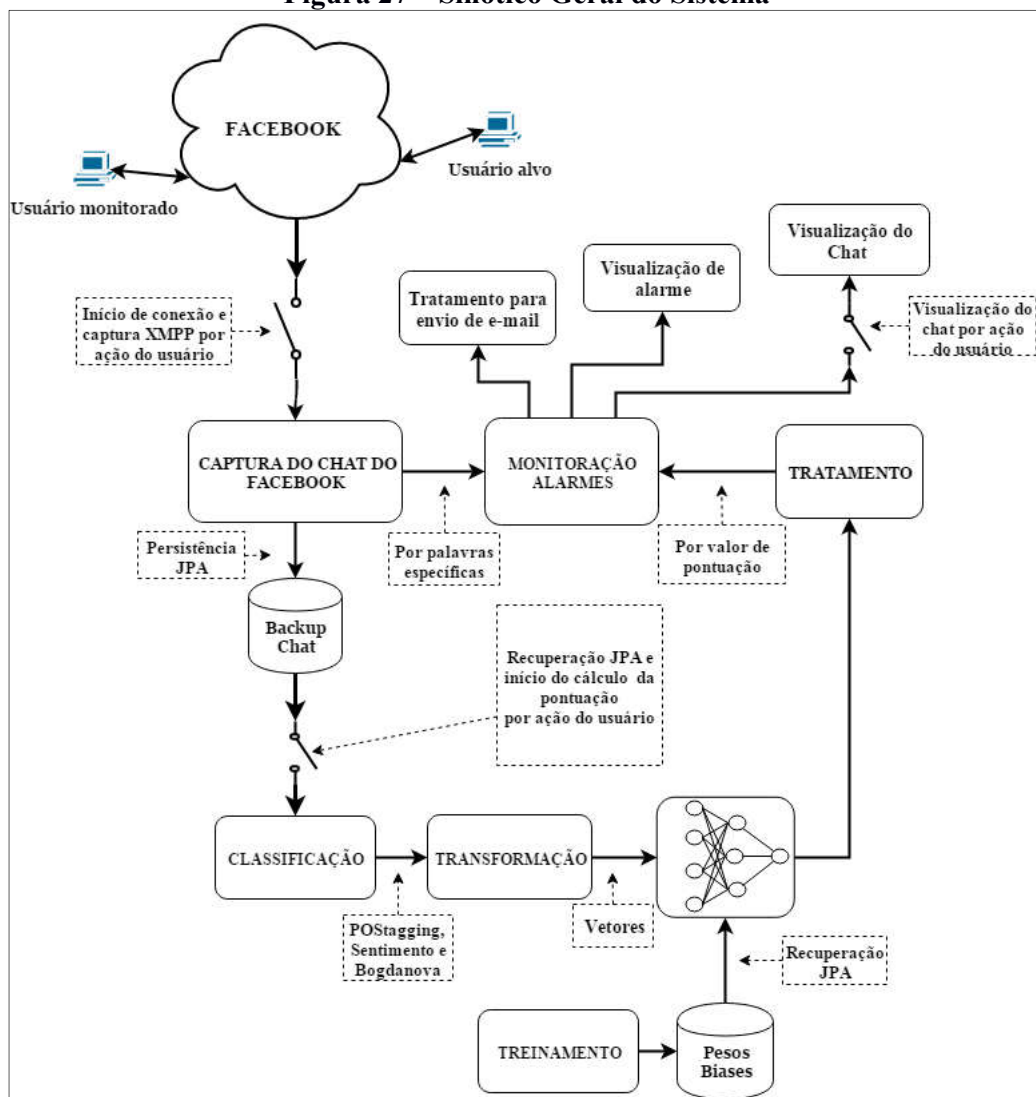
No caso de iniciada a conexão, métodos da biblioteca Smack são utilizados para a decodificação das mensagens XMPP provenientes do usuário remoto. Depois de decodificadas, as mensagens passam a ser monitoradas, persistidas no banco de dados e podem também ser visualizadas.

Por ação do usuário as mensagens podem ser recuperadas do BD para iniciar o restante do processo de detecção. Se assim o usuário proceder, as mensagens alimentam os estágios de classificação e transformação a fim de iniciar o cálculo de sentimento do texto completo recuperado do BD e a classificação das palavras com os métodos da biblioteca OpenNLP conforme o trabalho de Bogdanova (2012).

Com a classificação executada, no estágio seguinte a mensagem completa e etiquetada é transformada no vetor adequado para alimentar a rede neural que recebeu os pesos e polarizações que foram previamente calculados nas etapas de treinamento, validação e checagem.

O valor de saída calculado pela RNA é formatado para exibição e enviado ao estágio de monitoração de nível. Se algum alarme ocorrer por palavras ou frases específicas ou por nível de pontuação, envia-se uma mensagem a uma ou mais contas de e-mail pré-configuradas e também a um objeto da interface visual do sistema.

Figura 27 – Sinótico Geral do Sistema



Fonte: o autor

Nos subcapítulos seguintes são apresentados detalhes referentes à implementação do sistema.

3.2.1- Captura das Mensagens do Facebook

A captura das mensagens provenientes do chat do Facebook do usuário monitorado com destino ao usuário conhecido inicia-se com a aplicação de métodos da biblioteca Smack, os quais permitem o recebimento e a decodificação de mensagens.

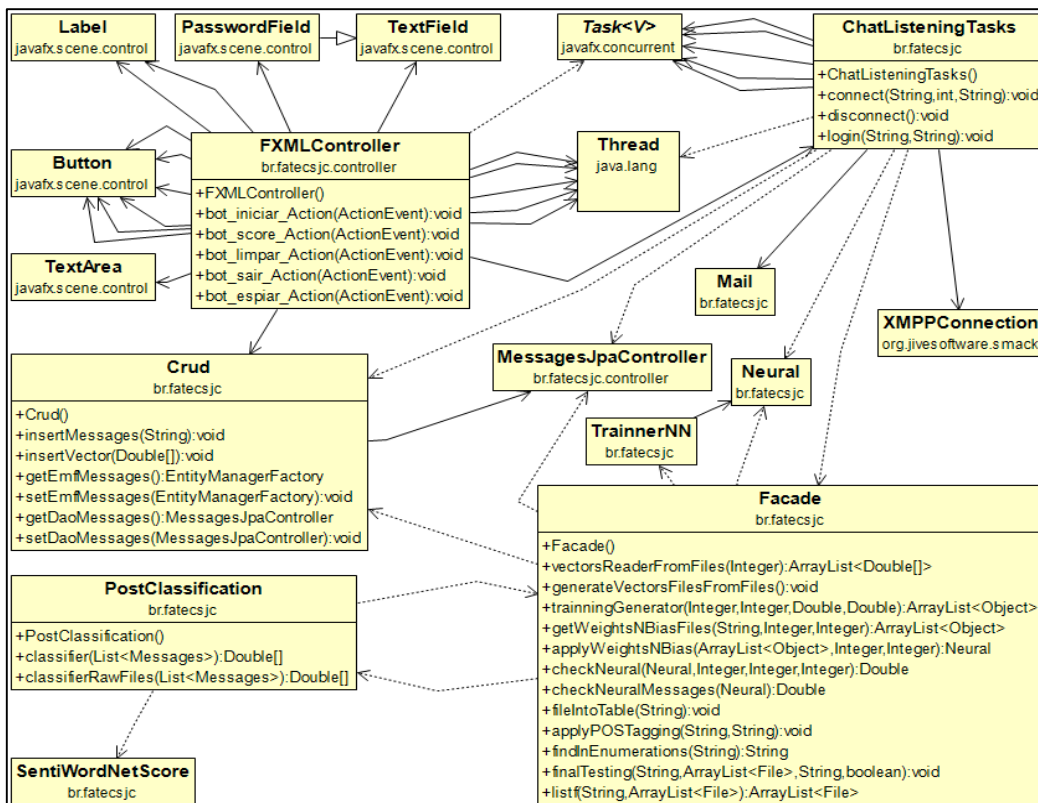
Não foi implementada funcionalidade alguma que permita separar mensagens oriundas de mais de um usuário monitorado. Sendo assim, o Chatlisten permite a monitoração de apenas um usuário por vez.

Com relação aos aspectos de segurança, a conexão com o Facebook usando o a biblioteca Smack com o protocolo XMPP é feita, por padrão, aplicando o protocolo TLS (*Transport Layer Security*) entre a camada aplicação e a camada de transporte. Assim, se for necessária uma conexão usando texto plano ou sem criptografia, deve-se alterar o código e ajustar as configurações da conexão por meio do seguinte método: `setSecurityMode(SecurityMode.disabled)`.

Na Figura 28 tem-se o um diagrama contendo as classes e os relacionamentos envolvidos do sistema completo. Para as classes principais constam o nome do pacote, o nome da classe e os métodos utilizados, porém, para fins de simplificação, nas demais classes constam apenas o nome da classe e o pacote ao qual ela pertence.

Cada ação do usuário é iniciada pressionando-se um botão e, para cada um destes botões, associa-se uma Thread e cada uma destas executa os métodos constantes de sua Task respectiva.

Figura 28 – Diagrama Geral do Chatlisten

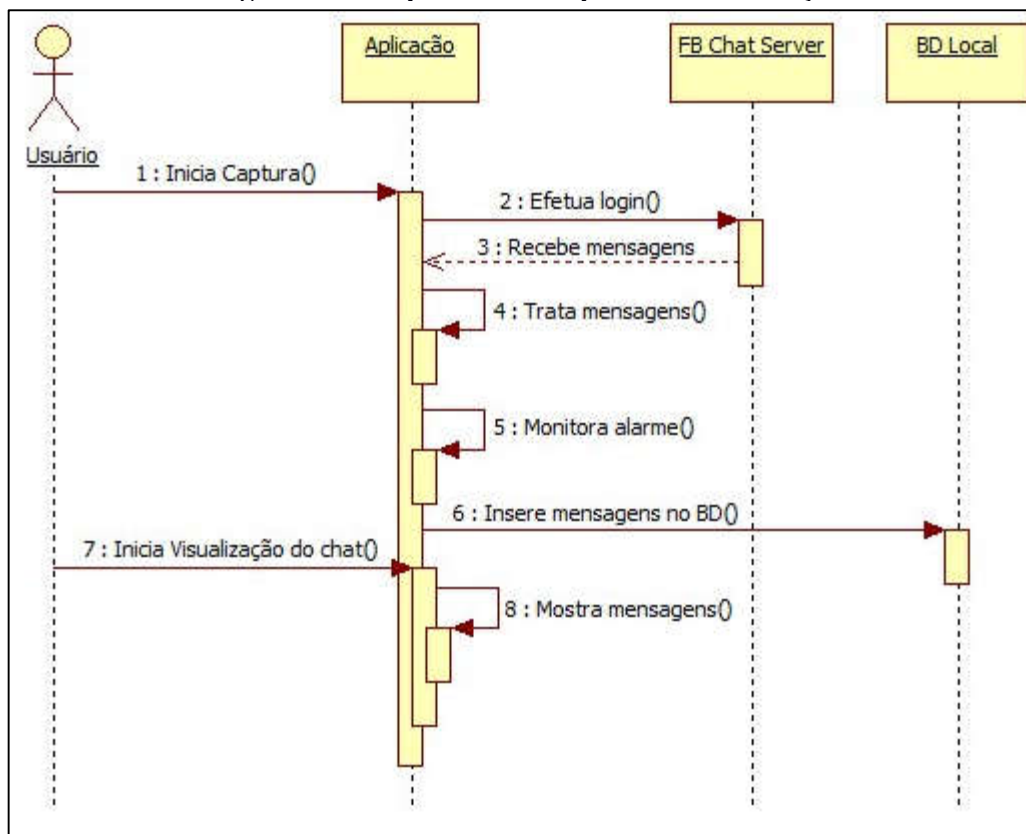


Fonte: o autor

Na Figura 29 é apresentado um diagrama de sequência no qual constam as principais etapas do processo de captura e monitoração das mensagens do chat do Facebook. Neste diagrama, toda codificação é representada no objeto Aplicação e as interações com os outros objetos (FB Chat Server e BD Local) são simplificadas.

O objeto Usuário representa a pessoa que conhece as credenciais de acesso (login e senha) do menor que se deseja monitorar e proteger.

Figura 29 – Sequência de Captura e Monitoração

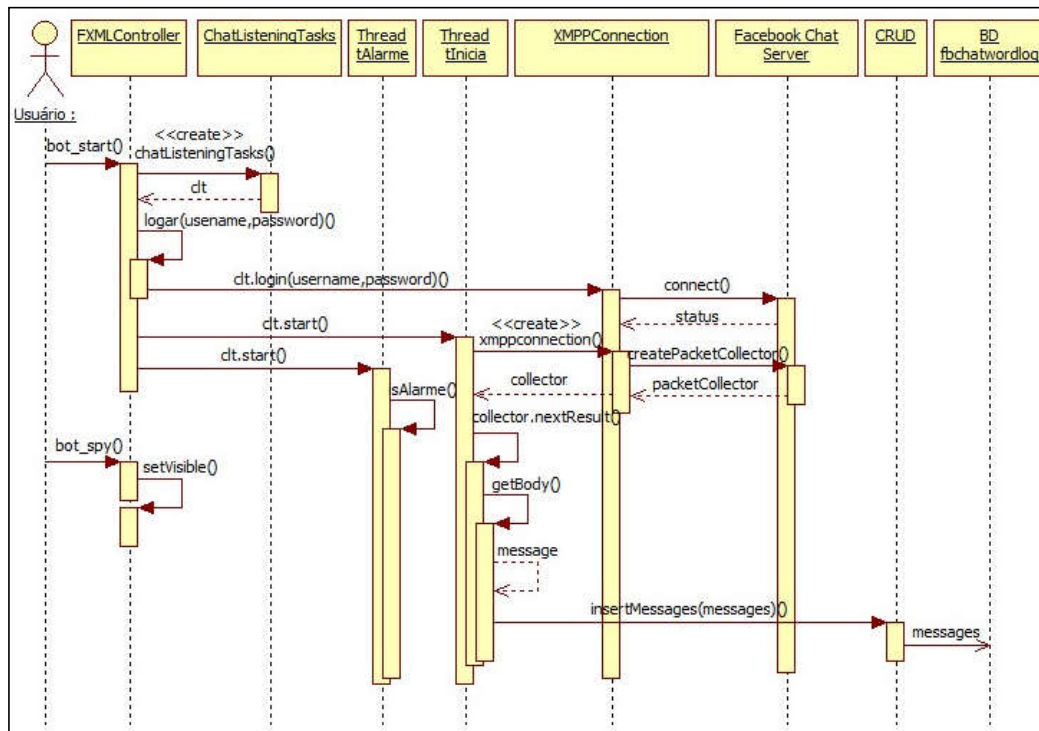


Fonte: o autor

A Figura 30 mostra o processo de captura e monitoração na forma de outro diagrama de sequência, porém com parte dos métodos reais utilizados na codificação do sistema. Percebe-se que o processo envolve o uso de *threads* para a captura das mensagens e para a monitoração do alarme de palavras ou frases específicas.

Algumas conversões de tipo e mesmo instanciações de algumas classes foram omitidas para que se pudesse simplificar e assim entender melhor a sequência do processo de inicialização da captura e monitoração.

Figura 30 – Detalhes da Sequência de Captura e Monitoração



Fonte: o autor

Segue na Figura 31 um trecho do código da *Task* *taskInicia* utilizado na captura das mensagens do chat do Facebook. Vê-se que, com a criação de poucos objetos da biblioteca Smack e a aplicação de alguns de seus métodos, tornou-se possível o recebimento de mensagens criadas segundo o protocolo XMPP.

Figura 31 – Captura das Mensagens

```

public Task<Void> taskInicia = new Task<Void>() {
    @Override
    protected Void call() throws Exception {
        PacketFilter filter = new AndFilter(new PacketTypeFilter(Message.class));
        PacketCollector collector = xmppConnection.createPacketCollector(filter);
        log = "";
        updateMessage("Listening...");
        while (!isCancelled()) {
            Packet packet = collector.nextResult();
            if (packet instanceof Message) {
                Message message = (Message) packet;
                String msg = message.getBody();
            }
        }
    }
}
  
```

Fonte: o autor

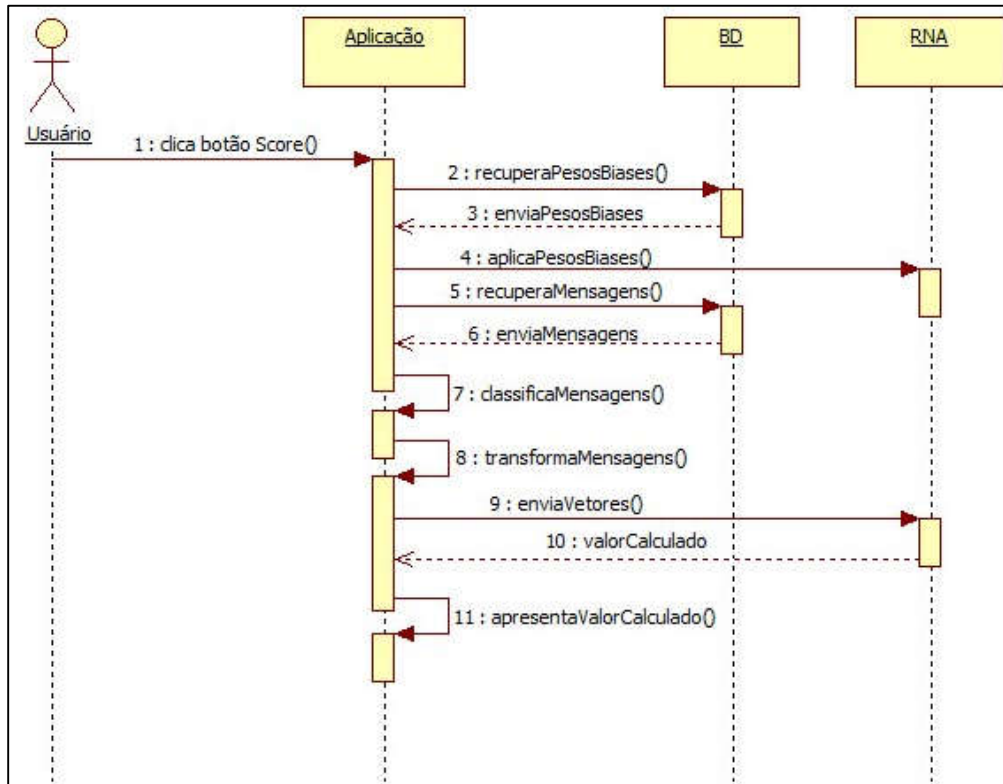
3.2.2- Cálculo da Pontuação

Para que se procedesse ao cálculo da pontuação do usuário monitorado, que possibilita avaliar se ele é ou não um predador sexual, três métodos da classe Facade

foram utilizados para criar uma lista com os pesos e as polarizações da RNA com menor saída, passar essa lista para a rede neural e efetuar o cálculo da pontuação.

Na Figura 32 tem-se um diagrama mostrando a sequência do processo disparado quando o usuário pressiona o botão Score. O objeto Aplicação representa todo o código do sistema de forma similar ao processo de captura de mensagens.

Figura 32 – Sequência do Cálculo da Pontuação



Fonte: o autor

O processo inicia-se com a recuperação e aplicação das configurações da rede neural. Em seguida as mensagens são lidas do banco de dados para que se dê início às fases de classificação e transformação. Nestas fases as mensagens são classificadas com base no recurso léxico SentiWordNet e no trabalho de Bogdanova (2012) e são transformadas nos vetores adequados à entrada da RNA. A rede neural, já com os ajustes aplicados, calcula o valor com base no vetor de entrada, liberando para apresentação a saída calculada.

Na Figura 33 segue um trecho do código da *task* taskScore. A referida *task* é passada como argumento na inicialização da *thread* tScore sempre que o usuário clicar no botão Score. O código permite buscar a lista de mensagens já persistidas no

BD por meio de métodos da classe Crud. Vê-se que para o real início do cálculo é necessária uma quantidade mínima arbitrada de 30 palavras.

Tendo-se uma lista com quantidade maior que 30, procede-se à busca e aplicação dos pesos e polarizações na RNA. Na sequência o valor de saída é calculado com a passagem do objeto *neural* ao método *checkNeuralMessages* da classe Facade. Então, o valor calculado pela rede neural é formatado e enviado ao objeto *label* da interface visual.

De acordo com o resultado da avaliação apresentada no subcapítulo 3.1.5 o valor de 0,4 é ajustado como limiar de início de alarme de nível de pontuação, sendo que duas *threads* permanecem monitorando o nível de pontuação e, por meio de alteração nas cores de fundo do objeto *label* e envio de mensagens a outro objeto *label*, funcionam como um indicador visual de alarme. Assim quanto mais próximo do vermelho a cor de fundo do *label* está, mais alta é a pontuação e uma mensagem de envio de e-mail é apresentada.

Figura 33 – Cálculo da Pontuação

```
public Task<Void> taskScore = new Task<Void>() {
    @Override
    protected Void call() throws Exception {
        DecimalFormat df = new DecimalFormat("0.000");
        while (!isCancelled()) {
            Facade facade = new Facade(); Crud crud = new Crud(); int tamanho = 1;
            List<Messages> listMsg = crud.daoMessages.findMessagesEntities();
            for (int i = 0; i < listMsg.size(); i++) {
                String line = listMsg.get(i).getLinha();
                String[] words = line.split(" "); tamanho += words.length;
            }
            if (tamanho < 30) {
                updateMessage("Poucas palavras - " + tamanho);
                cancelled(); break;
            }
            ArrayList<Object> weightsNBias = facade.getWeightsNBiasFiles("Output", 17, 5);
            score = facade.checkNeuralMessages(facade.applyWeightsNBias(weightsNBias, 17, 5));
            updateMessage("");
            updateMessage(df.format(score));
            if (score > 0.4) {
                alarmeScoreEstilo = true; alarmeScoreTexto = true;
                cancelled();
                break;
            }
            Integer minuto = 60000;
            Thread.currentThread();
            Thread.sleep(10*minuto);
        }
        Thread.currentThread().interrupt();
    }
}
```

Fonte: o autor

Na Figura 34 é apresentada a tela do sistema Chatlisten, criada com o Scene Builder. Nela existem dois objetos *textbox*, sendo um do tipo *password* e um do tipo *textArea*, cinco objetos *labels*, sendo um para apresentação do cálculo da pontuação, um para mensagens de erro e avisos e os demais são usados como descrições de campos e, finalmente, cinco objetos *buttons* para disparo de eventos.

Figura 34 – Tela do Sistema Chatlisten



The screenshot shows a window titled "ChatListen" with standard Windows window controls (minimize, maximize, close). The interface is divided into two main sections. On the left, there is a large rectangular area labeled "Incoming Messages". On the right, there is a vertical stack of elements: at the top, two input fields labeled "User" and "Password" with placeholder text "Enter username" and "Enter password" respectively; below these, a box labeled "Warnings"; then a box labeled "Score"; and at the bottom, a row of buttons. This row includes a button labeled "Score" on the left, and a group of three buttons labeled "Start", "Hide", "Clean", and "Exit" on the right.

Fonte: o autor

4- ANÁLISE E DISCUSSÃO DOS RESULTADOS

Neste capítulo são confrontados os resultados alcançados com os objetivos e requisitos apresentados no capítulo 1. Um dos objetivos apresentados na introdução deste trabalho de graduação – criar e implementar um sistema para monitoração em tempo real do mecanismo de chat do Facebook – foi integralmente alcançado. Com relação à capacidade de detecção de predadores sexuais, considera-se um objetivo parcialmente alcançado.

Nos subcapítulos 4.1, 4.2 e 4.3, são apresentados respectivamente os testes aplicados ao sistema, os resultados obtidos e, por fim, um comparativo entre as soluções existentes e a solução proposta deste trabalho de graduação.

4.1- Teste de Detecção

Os testes realizados para detecção de predadores foram efetuados usando-se arquivos de *log* do tipo predador do PJ, não predador da NPS e do CybeSex e também vetores já criados com os métodos de classificação e transformação suscitados no capítulo 3.1.3. Esse conjunto de *logs* foi reservado unicamente para testes e foi composto por 10 arquivos escolhidos de modo aleatório dentre os arquivos disponíveis do PJ e do CyberSex. Da NPS foram usados 5 arquivos, formando assim um conjunto de 25 arquivos.

De cada arquivo foi copiado um trecho de 50 linhas e colado na caixa de texto de envio do chat do Facebook de um usuário criado para testes. O texto foi enviado e recebido normalmente pela aplicação Chatlisten. Na aplicação foi utilizada outra conta do Facebook a fim de simular o usuário conhecido – aquele cujas credenciais são conhecidas e se deseja monitorar e vigiar.

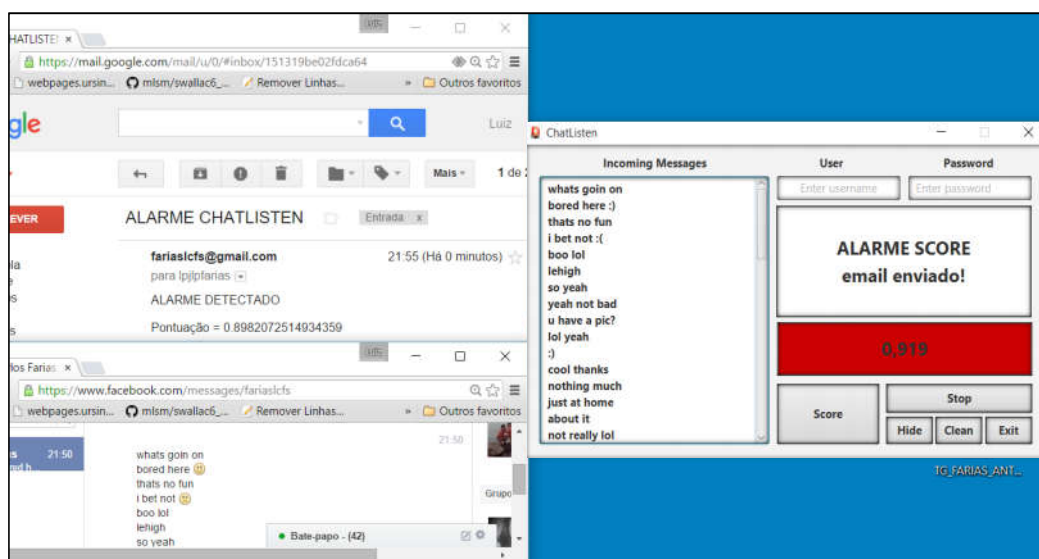
Para os vetores, foi criada uma lista contendo os vetores correspondentes ao conjunto de teste de *logs*. Usando-se os vetores diretamente, as etapas de seleção, limpeza e transformação são transpostas e o processo torna-se mais rápido. Pode-se também, por meio do uso direto de vetores, testar as referidas etapas iniciais de KDD.

O rótulo que apresenta o valor da pontuação tem a cor de fundo alterada de acordo com faixas de valores de pontuação. Acima de 0,5 a cor de fundo (*background color*) varia do amarelo até o vermelho (maior que 0,93) segundo uma escala simplificada similar à de indicação visual de calor.

Para o teste de detecção de palavra ou frase específica PFE foi inserida diretamente no código a sigla FATECSJC como palavra ou frase específica a monitorar. Assim, se a referida palavra fosse recebida pela aplicação, um alarme específico seria acionado e um e-mail seria enviado para a conta configurada no código fonte.

Na Figura 35 tem-se a pontuação calculada para o primeiro arquivo de *log* do tipo predador. O valor calculado foi de 0,918602080952661, porém para apresentação na interface, a saída foi formatada para três casas decimais. Tem-se também o navegador com duas janelas abertas, mostrando em uma a caixa de entrada do Gmail com o e-mail de alerta recebido e outra janela com o chat do Facebook mostrando um chat reproduzido do PJ que foi enviado para teste.

Figura 35 – Pontuação com Sistema Completo e Log do PJ

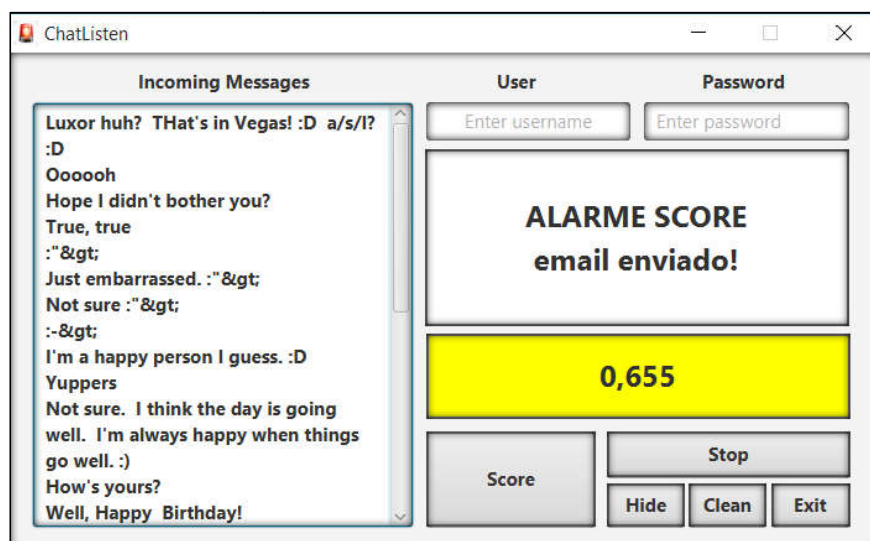


Fonte: o autor

Na Figura 36 o Chatlisten apresenta o resultado calculado para outro *log* do tipo predador, porém com um valor menor mostrando assim um alarme visual de outra cor.

Com relação à cor de fundo do rótulo de alarme, foram definidas faixas de cores de modo a simular uma escala similar à escala de calor. Assim, quanto maior a pontuação, mais próximo do vermelho será a cor de fundo do rótulo. No sentido oposto, quanto menor a pontuação, mais próximo do verde será a cor.

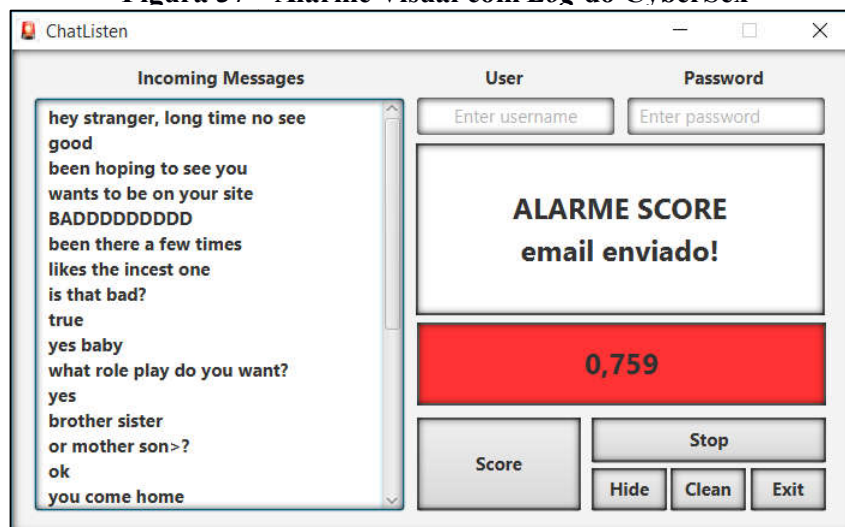
Figura 36 – Alarme Visual com Log do PJ



Fonte: o autor

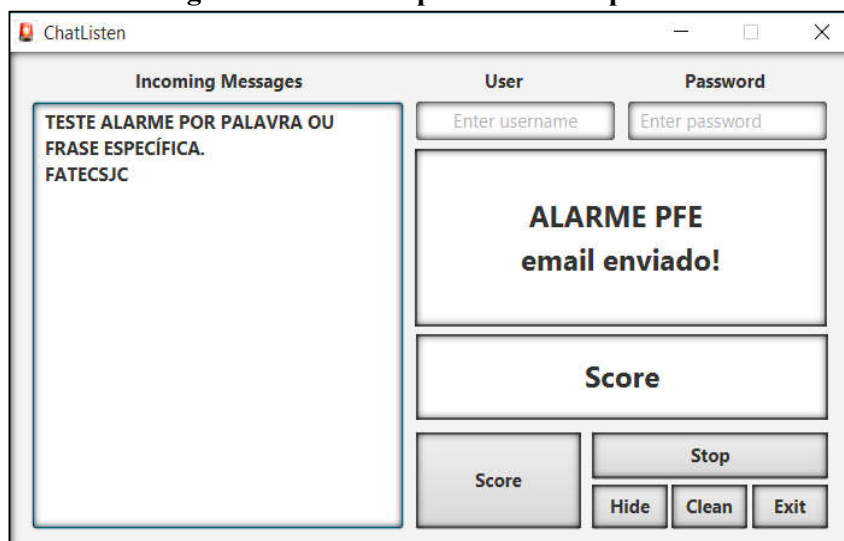
A Figura 37 apresenta um teste efetuado com log do CyberSex.

Figura 37 – Alarme Visual com Log do CyberSex



Fonte: o autor

Na Figura 38 é apresentado o Chatlisten com alarme PFE (Palavras ou Frases Específicas). Este alarme refere-se à monitoração de uma sequência de caracteres embutida no código fonte da taskInicia. Este alarme foi criado, primeiramente, para testes, porém mostrou-se muito útil para fins de monitoração sobre o conteúdo das mensagens.

Figura 38 – Alarme por Palavra Específica

Fonte: o autor

4.2- Análise dos Resultados

Nas Tabelas 5, 6 e 7 são apresentados os valores de saída, esperado e erro para cada tipo de *log* do conjunto de teste. Esses valores foram gerados pelo processo que envolve o acesso ao BD em busca da mensagem até o cálculo de saída usando-se a interface visual.

Tabela 5 – Teste com Logs do PJ

LOG	SAÍDA	VALOR ESPERADO	ERRO
1	0,9186	1,0	0,0814
2	0,2857	1,0	0,7143
3	0,9967	1,0	0,0033
4	0,6548	1,0	0,3452
5	0,9711	1,0	0,0289
6	0,6248	1,0	0,3752
7	0,0471	1,0	0,9529
8	0,9888	1,0	0,0112
9	0,9821	1,0	0,0179
10	0,1746	1,0	0,8254
EMQ			0,2367

Fonte: o autor

Tabela 6 – Teste com *Logs* da NPS

LOG	SAÍDA	VALOR ESPERADO	ERRO
1	0,2877	0,0	-0,2877
2	0,0867	0,0	-0,0867
3	0,3050	0,0	-0,3050
4	0,1507	0,0	-0,1507
5	0,1990	0,0	-0,1990
6			0,0000
7			0,0000
8			0,0000
9			0,0000
10			0,0000
EMQ			0,0246

Fonte: o autor

Tabela 7 - Teste com *Logs* do CyberSex

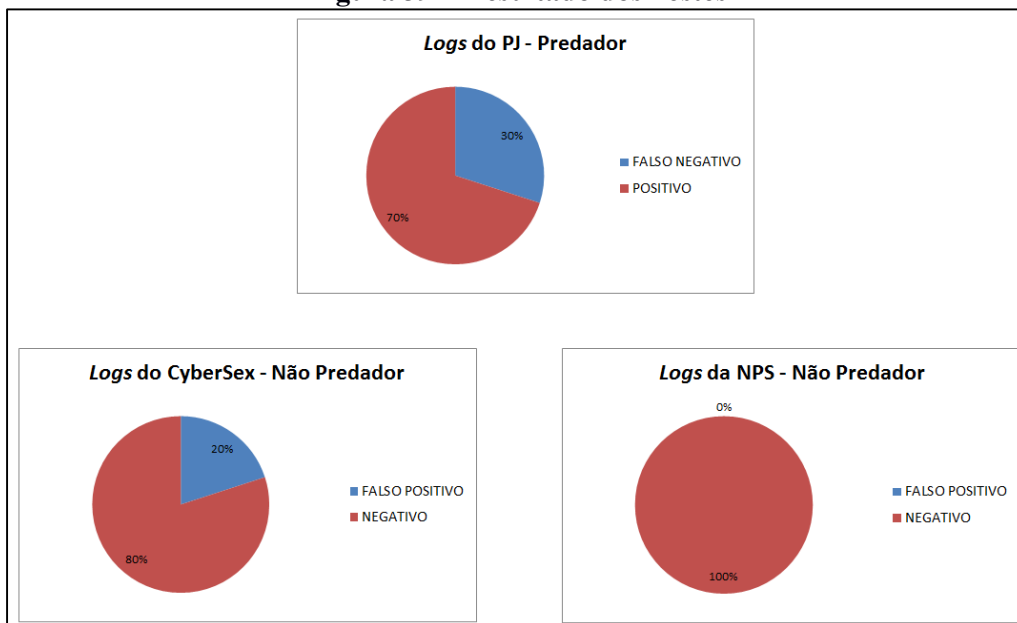
LOG	SAÍDA	VALOR ESPERADO	ERRO
1	0,1028	0,0	-0,1028
2	0,0364	0,0	-0,0364
3	0,7589	0,0	-0,7589
4	0,1293	0,0	-0,1293
5	0,0698	0,0	-0,0698
6	0,5049	0,0	-0,5049
7	0,0685	0,0	-0,0685
8	0,0709	0,0	-0,0709
9	0,1363	0,0	-0,1363
10	0,2436	0,0	-0,2436
EMQ			0,0952

Fonte: o autor

Com base nas Tabelas 5, 6 e 7, pode-se notar que considerando o ponto arbitrado de 0,4 os testes com *logs* do tipo predador classificaram corretamente 70% do total aplicado. Os *logs* da NPS foram todos classificados corretamente e os do CyberSex obtiveram 80% de acertos. A Figura 39 apresenta esses resultados em forma de gráfico para melhor visualização.

Com relação aos *logs* do PJ, o sistema classificou corretamente 7 dos 10 *logs* testados. Foram 7 acertos (Positivos) e 3 erros (Falsos Negativos). Sobre os *logs* do CyberSex, o sistema classificou corretamente 8 dos 10 *logs* testados. Foram 8 acertos (Negativos) e 2 erros (Falsos Positivos). Para os *logs* da NPS, o sistema classificou corretamente os 5 *logs* testados. Foram 5 acertos (Negativos) e nenhum erro (Falso Positivo).

Figura 39 – Resultado dos Testes



Fonte: o autor

4.3- Comparativo entre Soluções Existentes e Sistema Proposto

A Tabela 8 apresenta um comparativo entre as soluções existentes e a solução proposta neste trabalho por meio de um cruzamento entre o título das soluções e as principais características de cada uma.

De todas as soluções encontradas, somente duas apresentam-se como software, sendo que nenhuma destas possui monitoração em tempo real e tem por objetivo principal a detecção de assediadores sexuais.

Tabela 8 – Cruzamento Título / Características

TÍTULO	MONITORA ASSÉDIO SEXUAL	MONITORAÇÃO EM TEMPO REAL	ANÁLISE DE SENTIMENTOS	SOFTWARE	ALARME REMOTO
CHATCODER	X	X	X	✓	X
CHATLISTEN	✓	✓	✓	✓	✓
A Two-step Approach for Effective Detection of Misbehaving Users in Chats	✓	X	✓	X	X
Análise Automática de Textos de Mensagens Instantâneas para Detecção de Aliciamento Sexual de Crianças e Adolescentes	✓	X	✓	X	X
BULLYTRACKER Detecting the Presence of Cyberbullying Using Computer <i>Software</i>	X	X	X	✓	X

Fonte: o autor

5- CONCLUSÃO

Neste capítulo final, são apresentadas as contribuições deste Trabalho, as conclusões tiradas do desenvolvimento e são também mostradas algumas sugestões para trabalhos futuros, tudo com a finalidade contribuir com a contínua pesquisa científica, permitindo assim que os acertos obtidos com este trabalho sejam aplicados e aprimorados e que os erros ou insucessos sirvam como pontos a evitar ou mesmo corrigir.

5.1- Contribuições

São lançadas aqui as contribuições oferecidas com o desenvolvimento do corrente trabalho com base nos requisitos apresentados no capítulo 1.

Pesquisa Bibliográfica

A pesquisa bibliográfica, antes de quaisquer sucessos ou insucessos, é, por si só, uma contribuição natural à pesquisa científica e este trabalho de graduação é uma rica fonte de pesquisa no que tange ao assunto proposto, quer seja para desenvolvimento de outros sistemas ou, simplesmente, para consulta ou aprendizado.

Monitoração do chat do Facebook

A ferramenta proposta permite a monitoração não invasiva das mensagens do chat do Facebook de um usuário já conectado. Assim, contribui para a vigilância das crianças e adolescentes que usam o referido chat.

Cálculo de Pontuação em chats e Análise de Sentimentos

Ainda que não completamente implementado o método de Bogdanova (2012), tem-se um sistema de *software* criado para calcular a pontuação referente à possibilidade de haver um predador usando um determinado chat. Além da implementação parcial do referido método, usa-se, também, análise de sentimentos com o SentiWordNet.

Assim, a ferramenta criada contribui para contínua busca na criação de um sistema definitivo de detecção de predadores sexuais nas redes sociais.

Monitoração e Vigilância de Crianças e Adolescentes

A funcionalidade de envio de alarmes via e-mail contribui com a monitoração e vigilância constantes dos jovens e crianças que usam o chat do Facebook como uma das formas de comunicação existentes atualmente.

Testes com RNA Usando Vetores Gerados a partir de Textos

Tendo em vista o desbalanceamento na quantidade de *logs* aplicados em todas as fases (treinamento, validação e testes) e os resultados alcançados nos testes finais com o uso da interface visual, observa-se a viabilidade do uso de redes neurais artificiais do tipo MLP para detecção de assediadores sexuais na Internet.

5.2- Conclusão Geral

Considerando os resultados obtidos com o desenvolvimento deste trabalho e aqui apresentados conclui-se que a solução permite a monitoração do chat do Facebook de um usuário conectado e autenticado.

5.3- Sugestões para Trabalhos Futuros

Nesta última parte são oferecidas algumas sugestões para a continuação e aprimoramento do presente trabalho.

Eleva Quantidade de *Logs* de Não Predadores

É essencial que se tenha uma fonte de onde tirar os padrões de conversação dos predadores sexuais e também dos não predadores sexuais. Neste trabalho, houve um desbalanceamento quanto à quantidade de *logs*. Então, sugere-se que se busque por mais *logs* de não predadores.

Aprimorar os Métodos de Teste

No caso de se usar RNA, recomenda-se criar novos testes de modo a aumentar a velocidade com que os resultados são calculados e apresentados. Sugere-se também reestruturar o modo ou momento em que as mensagens são persistidas no banco de dados.

Privacidade

Sugere-se aplicar mecanismos de criptografia a fim de cifrar as mensagens gravadas no banco de dados e, desse modo, proteger a privacidade dos usuários monitorados.

Migração para Plataforma Móvel

Sugere-se a migração do sistema para plataformas móveis a fim de permitir que o monitoramento não fique restrito a um computador do tipo *desktop*.

Integrar Alarmes via SMS

No corrente trabalho não foi integrado métodos de alarme via SMS. Assim, recomenda-se buscar *gateways* gratuitos ou, no caso de não haver serviços não cobrados, recomenda-se integrar métodos com escolhas sugeridas no próprio sistema ou com escolhas fornecidas pelo usuário.

Permitir Escolha de Diferentes Métodos de Classificação

Foi usado no desenvolvimento deste trabalho somente redes neurais artificiais para o cálculo da pontuação. É recomendável criar outros métodos de classificação e cálculo e permitir que o usuário os escolha.

Adaptar ou Criar Sistemas para o Português Brasileiro

Tendo em vista que o presente trabalho foi criado com base em *logs* de arquivos extraídos do PJ, do CyberSex e da NPS, estando assim todos na Língua Inglesa, recomenda-se buscar informações sobre armazenamento de *logs* na Língua Portuguesa para, então, adaptar o corrente trabalho ou mesmo criar novos projetos que permitam a monitoração e vigilância das crianças e adolescentes que usam chats no Português Brasileiro.

REFERÊNCIAS

- AGUIAR, Sonia. **Redes sociais na internet: desafios à pesquisa**. Trabalho apresentado, 2007.
- BASHER, Abdur Rahman MA; FUNG, Benjamin CM. **Analyzing topics and authors in chat logs for crime investigation**. *Knowledge and information systems*, v. 39, n. 2, p. 351-381, 2014.
- BATES, Madeleine. **Models of natural language understanding**. *Proceedings of the National Academy of Sciences*, v. 92, n. 22, p. 9977-9982, 1995.
- BAYZICK, Jennifer; KONTOSTATHIS, April; EDWARDS, Lynne. **Detecting the presence of cyberbullying using computer software**. 2011.
- BOGDANOVA, Dasha; ROSSO, Paolo; SOLORIO, Tamar. **On the impact of sentiment and emotion based features in detecting online sexual predators**. In: *Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis*. Association for Computational Linguistics, 2012. p. 110-118.
- CASSETARI, Ailton. O princípio da máxima entropia e a moderna teoria das carteiras. *Revista Brasileira de Finanças*, v. 1, n. 2, p. pp. 271-300, 2003.
- CAVALIERI FILHO, Sergio. **Programa de sociologia jurídica**. 2010.
- COPPIN, B. **Inteligência Artificial**. LTC, Rio de Janeiro, 2010.
- CARBONELL, Jaime G.; MICHALSKI, Ryszard S.; MITCHELL, Tom M. **An overview of machine learning**. In: *Machine learning*. Springer Berlin Heidelberg, 1983. p. 3-23.
- CARDON, André; MÜLLER, Daniel Nehme; NAVAUX, Philippe. **Introdução Às Redes Neurais Artificiais**. *Instituto de Informática. Universidade Federal do Rio Grande do Sul. Porto Alegre*, 1994.
- CASTRO, FERNANDO CÉSAR C.; MARIA, CF DE CASTRO. **Redes Neurais Artificiais (Capítulo 4)**. PUCRS-FENG-DEE - Mestrado em Engenharia Elétrica, 2010.
- Código Internacional de Doenças (CID). **Código F65.4**. Disponível em: http://www.datasus.gov.br/cid10/V2008/WebHelp/f60_f69.htm. Acesso em: 06/04/2015.
- comScore. **Distribuição de Tempo Em Redes Sociais no Brasil**. Disponível em: <http://www.comscore.com/por/Insights/Presentations-and-Whitepapers/2014/The-State-of-Social-Media-in-Brazil-and-the-Metrics-that-Really-Matter>. Acesso em 31/10/2015.
- CyberSex**. Disponível em: <http://geocities.com/urgrl21f/>. Acesso em: 08/11/2015.

DA SILVA, Camila Cortellete Pereira; PINTO, Daniela Devico Martins; MILANI, Rute Grossi. Pedofilia, **Quem a Comete? Um Estudo Bibliográfico do Perfil do Agressor**.

DE ÁVILA OTHERO, Gabriel. **Linguística computacional: uma breve introdução**. *Letras de hoje*, v. 41, n. 2, p. 341-351, 2006.

DO LAGO PEREIRA, Silvio. **Introdução à Inteligência Artificial**. Disponível em: <http://www.ime.usp.br/~slago/ia-1.pdf>. Acesso em: 10/03/2015.

DO LAGO PEREIRA, Silvio. **Processamento de Linguagem Natural**. Disponível em <http://www.ime.usp.br/~slago/IA-pln.pdf>. Acesso em: 10/03/2015.

Estudo de caso. **Use K-Nearest Neighbors (KNN) Classifier in Java**. Disponível em: <http://www.programcreek.com/2013/01/use-k-nearest-neighbors-knn-classifier-in-java>. Acesso em: 10/04/2015.

FAYYAD, Usama; PIATETSKY-SHAPIO, Gregory; SMYTH, Padhraic. **From data mining to knowledge discovery in databases**. *AI magazine*, v. 17, n. 3, p. 37, 1996.

FRANCISCANI, Juliana F. **Inteligência Artificial: Uma Abordagem sobre Algoritmos Genéticos**. *Revista Rumos – Administração e Desenvolvimento*. Volume VI, 2012.

GONZAGA, Yuri. **Uso do Facebook por crianças no Brasil é triplo da média mundial, diz estudo**. Disponível em: <http://www1.folha.uol.com.br/tec/2014/01/1401800-uso-do-facebook-por-criancas-no-brasil-e-triplo-da-media-mundial-diz-estudo.shtml>. Acesso em: 27/11/2015.

GROSSMANN JR, Helmuth. **Um sistema especialista para o auxílio ao diagnóstico de problemas em computadores utilizando raciocínio baseado em casos**. Florianópolis: Programa de Pós-Graduação em Ciência da Computação. UFSC, 2002.

JACOBS, Elana T. **Online Sexual Solicitation of Minors: An Analysis of the Average Predator, His Victims, What Is Being Done and Can Be Done To Decrease Occurrences of Victimization**. *Cardozo Pub.L. Pol'y & Ethics J.*, v. 10, p. 505-575, 2012.

KOHONEN, T. **The Self-Organizing Map**. *Proceedings of the IEEE*, V.78, n. 9, Sep. 1990.

KONTOSTATHIS, April. **ChatCoder: Toward the tracking and categorization of internet predators**. In: *PROC. TEXT MINING WORKSHOP 2009 HELD IN CONJUNCTION WITH THE NINTH SIAM INTERNATIONAL CONFERENCE ON DATA MINING (SDM 2009)*. SPARKS, NV. MAY 2009. 2009.

KONTOSTATHIS, April; EDWARDS, Lynne; LEATHERMAN, Amanda. **Text mining and cybercrime. Text Mining: Applications and Theory.** John Wiley & Sons, Ltd, Chichester, UK, 2010.

LIPPMAN, R. P. **An Introduction to Computing With Neural Nets.** IEEE ASSP Magazine, v. 3, n. 4, apr. 1987

LORENA, Ana Carolina; DE CARVALHO, André CPLF. **Uma introdução às support vector machines.** *Revista de Informática Teórica e Aplicada*, v. 14, n. 2, p. 43-67, 2007.

MARTINS, Adriano. **Fundamentos de Computação Nuvem para Governos. Serviço Federal de Processamento de Dados (SERPRO)**, Brasília-DF, Brasil, 2010.

MCGHEE, India et al. **Learning to identify Internet sexual predation.** *International Journal of Electronic Commerce*, v. 15, n. 3, p. 103-122, 2011.

MONARD, Maria Carolina; BARANAUSKAS, José Augusto. **Conceitos sobre aprendizado de máquina.** *Sistemas Inteligentes-Fundamentos e Aplicações*, v. 1, p. 1, 2003.

MORAIS, Edison Andrade Martins; AMBRÓSIO, Ana Paula L. **Mineração de Textos.** *Instituto de Informática Universidade Federal de Goiás*, 2007.

MOREIRA, T. M. M., Viana, D. D. S., Queiroz, M. V. O., & Jorge, M. S. B. (2008). **Conflitos vivenciados pelas adolescentes com a descoberta da gravidez.** *Rev Esc Enferm USP*, 42(2), 312-20.

MOURA, M. A. (2005). **Interações Sociais e Comunidades Virtuais: transformações na sociabilidade?** *Informática Pública*, 7(1), 85-97.

O Cenário das Redes Sociais e Métricas que Realmente Importam. Disponível em: <http://www.comscore.com/por/Insights/Presentations-and-Whitepapers/2014/The-State-of-Social-Media-in-Brazil-and-the-Metrics-that-Really-Matter>. Acesso em: 17/04/2015.

PAMPLONA FILHO, R. (2002). **Assédio sexual: questões conceituais.** *Assédio Sexual*. JESUS, Damásio Evangelista de; GOMES, Luiz Flávio. (Coordenadores). São Paulo: Saraiva.

PAUCAR, Leonardo. Programa da disciplina "Inteligência Artificial", UFMA, 2000-1. Capítulo 4. **Sistemas Especialistas.** Disponível em: <http://www.dee.ufma.br/~lpaucar/teaching/ia2000-1/cap4.html>. Acesso em: 10/04/2015.

PENDAR, Nick. **Toward spotting the pedophile telling victim from predator in text chats.** In: null. IEEE, 2007. p. 235-241.

PBM 2015. **Pesquisa Brasileira de Mídia.** Disponível em: <http://www.secom.gov.br/atuacao/pesquisa/lista-de-pesquisas-quantitativas-e->

qualitativas-de-contratos-atuais/pesquisa-brasileira-de-midia-pbm-2015.pdf. Acesso em: 10/11/2015.

PJ. Disponível em: <http://www.perverted-justice.com/index.php?pg=faq#cat1>. Acesso em 10/04/2015.

Processo de Engenharia do Conhecimento. **Etapas de KDD.** Disponível em: http://www.egc.ufsc.br/wiki/index.php/Processo_de_Engenharia_do_Conhecimento. Acesso em 13/11/2015.

RAHMANMIAH, Md Waliur; YEARWOOD, John; KULKARNI, Sid. **Detection of child exploiting chats from a mixed chat dataset as a text classification task.** In: Proceedings of the Australian Language Technology Association Workshop. 2011. p. 157-165.

REZENDE, S. O. Pugliesi, J. B., Melanda, E. A., & Paula, M. D. (2003). Mineração de dados. Sistemas inteligentes: fundamentos e aplicações, 1, 307-335.

SAINT-ANDRE, Peter; SMITH, Kevin; TRONÇON, Remko. **XMPP: the definitive guide.** "O'Reilly Media, Inc.", 2009.

SALTER, A. **Predadores: pedófilos, estupradores e outros agressores sexuais.** São Paulo: Mbooks, 2009.

SANTIN, Priscila Louise Leyser. **Análise Automática de Textos de Mensagens Instantâneas para Detecção de Aliciamento sexual de Crianças e Adolescentes.** 2013. Tese de Mestrado. Pontifícia Universidade Católica do Paraná.

SIMPSON, P. K. **Artificial Neural Systems: Foundations, Paradigms, applications, and implementations.** Pergamon Press, 1990.

SCHILDT, Herbert. **Advanced turbo prolog.** Berkley: McGraw Hill, 1987.

HEMANN, Charles. **Turing Test Success Marks Milestone in Computing History.** Disponível em: <http://www.reading.uk/news-adn-events/releases/pr583836.aspx>. Acesso em 10/11/2015.

SOUZA, MSc Hugo Vieira L. **Introdução a Inteligência Artificial: histórico, aplicações, abordagens e problemas.** 2014.

STRAPPARAVA, Carlo et al. **WordNet Affect: an Affective Extension of WordNet.** In: LREC. 2004. p. 1083-1086.

SYDOW, Spencer Toth. **“Pedofilia virtual” e considerações críticas sobre a lei 11.829/08.** Revista Liberdade IBCCrim, n 1 – maio – agosto de 2009.

THOMÉ, Antonio Carlos Gay. **Redes neurais: uma ferramenta para KDD e data mining.** Material Didático. Disponível em: http://equipe.Nce.ufrj.br/thome/grad/nn/mat_didatico/apostila_kdd_mbi.pdf. Outubro, 2002. Acesso em 10/11/2015.

PJ.Disponível em:

http://www.perverted-justice.com/?archive=fleet_captain_jaime_wolfe.

Acesso em: 10/04/2015.

VIEIRA, Renata; LIMA, Vera LS. **Linguística computacional: princípios e aplicações.** In: Anais do XXI Congresso da SBC. I Jornada de Atualização em Inteligência Artificial. 2001. p. 47-86.

VILLATORO-TELLO, Esaú et al. **A Two-step Approach for Effective Detection of Misbehaving Users in Chats.** In: CLEF (Online Working Notes/Labs/Workshop). 2012.