UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE CIÊNCIAS EXATAS E DA TERRA
DEPARTAMENTO DE INFORMÁTICA E MATEMÁTICA APLICADA
PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS E COMPUTAÇÃO

Dissertação de Mestrado

Sistema de Agentes Poligínicos para Esteganálise de Imagens Digitais

Samuel Oliveira de Azevedo

Natal, agosto de 2007.

Divisão de Serviços Técnicos

Catalogação da Publicação na Fonte. UFRN / Biblioteca Central Zila Mamede

Azevedo, Samuel de.
Sistema de agentes poligínicos para esteganálise de imagens digitais / Samuel Oliveira de Azevedo. – Natal, RN, 2007.
67 f.

Orientador: Luiz Marcos Garcia Gonçalves.

Dissertação (Mestrado) – Universidade Federal do Rio Grande do Norte. Centro de Ciências Exatas e da Terra. Departamento de Informática e Matemática Aplicada. Programa de Pós-Graduação em Sistemas e Computação.

1. Imagem digital – Esteganálise – Dissertação. 2. Esteganografia – Dissertação. 3. Segmentação de imagens – Dissertação. 4. Criptologia – Dissertação. 5. Poliginia – Dissertação. 6. Sistemas multi-agentes – Dissertação. I. Gonçalves, Luiz Marcos Garcia. II. Universidade Federal do Rio Grande do Norte. III. Título.

RN/UF/BCZM CDU 621.397(043.3)

Sistema de Agentes Poligínicos para Esteganálise de Imagens Digitais

Samuel Oliveira de Azevedo

Dissertação submetida ao Programa de Pós-graduação em Sistemas e Computação do Departamento de Informática e Matemática Aplicada da Universidade Federal do Rio Grande do Norte, como requisito parcial para obtenção de título de Mestre em Ciências (M.Sc.).

Aprovada em 06 de agosto de 2007, pela Comissão Examinadora formada pelos seguintes membros:

Prof. DSc. Luiz Marcos Garcia Gonçalves (Orientador)
Departamento de Engenharia de Computação e Automação – UFRN

Prof. DSc. Benjamin Rene Callejas Bedregal
Departamento de Informática e Matemática Aplicada, UFRN

Prof. PhD Ricardo Cordeiro de Farias
COPPE, UFRJ

Natal, agosto de 2007.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE CIÊNCIAS EXATAS E DA TERRA
DEPARTAMENTO DE INFORMÁTICA E MATEMÁTICA APLICADA
PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS E COMPUTAÇÃO

Dissertação de Mestrado

Sistema de Agentes Poligínicos para Esteganálise de Imagens Digitais

Samuel Oliveira de Azevedo

Dissertação submetida ao Programa de Pós-graduação em Sistemas e Computação do Departamento de Informática e Matemática Aplicada da Universidade Federal do Rio Grande do Norte, como requisito parcial para obtenção de título de Mestre em Sistemas e Computação (MSc).

Orientador : Prof. Dr. Luiz Marcos Garcia Gonçalves

Natal, agosto de 2007.

Agradecimentos

Agradeço à Deus por me permitir realizar o sonho que sempre foi para mim realizar este trabalho nessa área que tanto desejava trabalhar. Agradeço à minha família: minha mãe, Itaci, pela paciência e por todos os seus esforços em prover tudo o que precisamos, e minhas irmãs Jemima, Raquel e Priscila, por agüentarem meus momentos de mau humor, causados pelo stress do dia à dia e desse trabalho.

Agradeço aos professores Luiz Marcos, pela orientação; Guido Lemos, pela confiança em me dar a coordenação de um de seus projetos; Bruno Motta e Anne Magaly, por me darem as idéias que se juntaram às minhas para a base deste trabalho.

Agradeço ao meu amigo Aquiles, e aos demais amigos e colegas do laboratório Natalnet-DCA pelas discussões produtivas.

Agradeço àqueles que de alguma forma me ajudaram, pois sem eles eu não teria obtido o mesmo resultado. Agradeço à todos aqueles que não obstruíram meu caminho nessa jornada, pois ao menos não atrapalharam. E também agradeço aos que tentaram obstruir meu caminho, pois se tornaram desafios que me proporcionaram diversas superações.

Abstract

In this work, we propose a multi agent system for digital image steganalysis, based on the poliginic bees model. Such approach aims to solve the problem of automatic steganalysis for digital media, with a case study on digital images. The system architecture was designed not only to detect if a file is suspicious of covering a hidden message, as well to extract the hidden message or information regarding it. Several experiments were performed whose results confirm a substantial enhancement (from 67% to 82% success rate) by using the multi-agent approach, fact not observed in traditional systems. An ongoing application using the technique is the detection of anomalies in digital data produced by sensors that capture brain emissions in little animals. The detection of such anomalies can be used to prove theories and evidences of imagery completion during sleep provided by the brain in visual cortex areas.

Keywords

Steganography, steganalysis, image segmentation, cryptology, poliginy, multiagent systems, coordination.

Resumo

Neste trabalho, propomos um sistema multi-agentes para esteganálise em imagens digitais, baseado na metáfora das abelhas poligínicas. Tal abordagem visa resolver o problema da esteganálise automática de mídias digitais, com estudo de caso para imagens digitais. A arquitetura do sistema foi projetada não só para detectar se um arquivo é ou não suspeito de possuir uma mensagem oculta em si, como também para extrair essa mensagem ou informações acerca dela. Foram realizados vários experimentos cujos resultados confirmam uma melhoria substancial (de 67% para 82% de acertos) com o uso da abordagem multi-agente, fato não observado em outros sistemas tradicionais. Uma aplicação atualmente em andamento com o uso da técnica é a detecção de anomalias em dados digitais produzidos por sensores que captam emissões cerebrais em pequenos animais. A detecção de tais anomalias pode ser usada para comprovar teorias e evidencias de complementação do imageamento durante o sono, provida pelo cérebro nas áreas visuais do córtex cerebral.

Palavras Chave

Esteganografia, esteganálise, segmentação de imagens, criptologia, poliginia, sistemas multi-agentes, coordenação.

Sumário

Lista de Equações

Lista de Figuras

Lista de Tabelas

Capítulo 1.

Introdução

"A informação que temos não é a que desejamos. A informação que desejamos não é a que precisamos. A informação que precisamos não está disponível." John Peers

Certas informações demandam o uso de mecanismos de segurança para a sua transmissão. Entre as abordagens utilizadas para segurança de informação, encontra-se a criptografia e a esteganografia (Singh 2001).

A criptografia digital tem sido uma solução muito utilizada para diversos fins, dentre eles o comércio eletrônico (Luciano 2003), o voto eletrônico (Kofler 2003), e até pra TV digital (Macq 1995). Porém, um interceptor, monitorando o tráfego da rede, pode detectar facilmente a presença de uma mensagem criptografada. Em algumas ocasiões, esse interceptor, dispondo de recursos de processamento e tempo suficiente, pode até mesmo conseguir quebrar o código e descobrir o conteúdo da mensagem protegida.

Esse problema pode ser solucionado através do uso da esteganografia, que visa ocultar a existência das informações que se deseja proteger utilizando-se de outras informações como disfarce. Qualquer dado serve para ser utilizado como cobertura de esteganografia digital, tal como texto, áudio, vídeo, pacotes de rede, e sistemas de arquivo, entre outros. Entretanto, as imagens digitais são o meio de cobertura mais conhecido e talvez mais utilizado.

O que difere as duas abordagens é a ocultação da existência da informação, presente na esteganografia e a transformação dos dados que os torna ilegíveis a interceptores, presente na criptografia. A esteganografia de dados criptografados se torna uma ferramenta de segurança ainda mais poderosa.

Embora não se negue que todos têm a prerrogativa de proteger suas informações, em algumas situações, torna-se necessário que estas informações sejam reveladas para o bem maior da sociedade. Um exemplo de situação em que a quebra de sigilo é importante é na investigação criminal. Esteganálise é uma sub-área da criptologia que se

preocupa em detectar, e em alguns casos revelar o conteúdo de uma mensagem esteganografada.

Em resposta ao surgimento de novas técnicas de criptografia ou esteganografia surgem novas técnicas de criptoanálise ou esteganálise. A criptologia digital é um campo da segurança de informação que engloba tanto criptografia e criptoanálise quanto esteganografia e esteganálise. Uma verdadeira corrida é travada entre criptógrafos e criptoanalistas, e também entre esteganógrafos e esteganalistas: cada avanço de um dos lados pressiona o outro a superá-lo (Singh 2001).

Pela natureza do problema, que requer a esteganálise de grandes volumes de dados, torna-se mais adequado que tais análises sejam desempenhadas de forma autônoma por um sistema computacional. Outra característica do problema é que à medida que novas técnicas de esteganálise são desenvolvidas, surgem novas técnicas de esteganografia imunes às análises existentes. Por isso, mais uma característica que os sistemas de esteganálise demandam é a flexibilidade para adaptação às novas esteganografias. Esta flexibilidade pode ser obtida através de aprendizado ou do uso de técnicas de engenharia de software que facilitem a alteração do sistema em tempo hábil (como a modularização, documentação de código, etc).

Autonomia e flexibilidade para adaptação, são características comumente presentes em entidades de software chamadas de Agentes; e pela complexidade do problema, a abordagem de agentes para resolvê-lo é mais apropriada se utilizado um Sistema Multi-Agentes (SMA), que é um sistema onde diversos agentes especializados ou redundantes interagem (cooperando, negociando, trocando informações) para atingir seus objetivos(Garcia 2003).

Comumente, utilizamos como heurísticas algumas metáforas da natureza para resolver problemas computacionais de forma mais simples. Os SMAs são sistemas que abordam as interações sociais entre os agentes, portanto, torna-se necessário modelar ou projetar de alguma forma como se darão tais interações. Uma boa heurística nesse caso pode ser inspirada nas interações presentes nas sociedades de insetos, por exemplo.

Os insetos sociais apresentam características importantes como cooperação, divisão de trabalho, execução de múltiplas tarefas, e coordenação; que também são presentes em SMAs, onde os agentes (dentre outras atividades) interagem entre si. Este trabalho se inspira nas sociedades poligínicas das abelhas da espécie *Melipona Bicolor*

(Aponte 2003), onde as diversas rainhas que a colméia pode possuir cooperam na coordenação de todas as operárias. Estudamos tal modelo de coordenação e empregamos em nosso SMA.

Portanto, propomos neste trabalho um sistema multi-agentes de esteganálise, baseado na metáfora das abelhas poligínicas, para arquivos digitais; com um estudo de caso focado para as imagens digitais.

1.1. O Problema da Esteganálise Digital

Abordamos neste trabalho o problema de esteganálise de dados digitais. Como citado, tal atividade concorre com a esteganografia digital, ambos tentando superar um ao outro. Portanto, os métodos esteganalíticos devem ser adaptáveis às mudanças dos métodos esteganográficos, para que consigam ser capazes de detectar e até extrair os dados protegidos pela esteganografia. Um caso de uso de esteganálise é o monitoramento de tráfego numa rede em busca de dados ocultos, ocasionando situações em que o volume de dados a se analisar é grande (principalmente em redes com alto tráfego de dados). Por isso surge também a demanda de que o sistema esteganalítico atue de forma automática ou até mesmo autônoma sobre esses dados (Gonzalez 2002).

Essa esteganálise também deve ser feita com algum grau de confiabilidade respeitável. Com isso evita-se a detecção de ocorrências de dados ocultos inexistentes, evitando também que gastos desnecessários sejam efetuados ou conclusões equivocadas sejam tomadas, na tentativa de se extrair informação oculta onde não existe.

Caso uma fonte de informação, como por exemplo, um site ou um repositório de dados, seja examinada e a esteganálise não revele nenhum dado oculto, a confiabilidade da técnica de detecção empregada também pode servir como grau de garantia de que essa fonte está livre de suspeitas de estar servindo como veículo de comunicação esteganografada.

1.2. Contribuições Esperadas

Neste trabalho, propomos uma abordagem para esteganálise que visa detectar se um arquivo digital é ou não suspeito de possuir uma mensagem oculta em si. A estrutura do sistema está projetada para futuramente concluir o processo de esteganálise extraindo os dados protegidos. Para tanto, precisamos investigar técnicas de ataque à esteganografia, que detectem e que extraiam os dados secretos. A partir daí, propomos uma arquitetura flexível e autônoma o suficiente para atender aos requisitos de manipulação de grandes volumes de dados e adaptação às evoluções constantes das áreas envolvidas. Como citado anteriormente, uma solução que pode atender os requisitos de autonomia e flexibilidade é o uso de sistemas multi-agentes. Portanto, no presente trabalho desenvolvemos um SMA cuja estrutura seja adequada para a resolução do problema. Utilizaremos a heurística das abelhas poligínicas (mais detalhes no 2º. Capítulo) para desenvolver o modelo de coordenação dos agentes. Para simplificar a solução do problema, devido à complexidade do assunto, abordamos inicialmente o problema de forma mais focada, restringindo-nos apenas à esteganálise de imagens digitais. Como resultado, desenvolvemos uma arquitetura de sistema esteganalítico universal, que abranja todos os tipos de dados e técnicas esteganográficas. Assim, como resultado do presente trabalho, podemos ressaltar as seguintes contribuições principais:

- uma arquitetura autônoma e adaptável de sistema esteganalítico que seja universal, capaz de concorrer com ou superar os métodos de esteganálise e esteganografia atuais;
- uma nova heurística de coordenação de sistemas multi-agentes.

1.3. Justificativa e Aplicações

A esteganálise, como todo assunto da área de segurança, é importante na definição de estratégias para a economia, defesa e até a soberania nacional (Alves 2002). Dominar as tecnologias de segurança é um passo importante visando sua utilização de forma adequada para proteger informações importantes. Compreender a esteganálise é também compreender melhor os meios de esteganografia correntes, uma vez que são fortemente correlacionadas. Além do uso comercial, a esteganálise também pode ser aplicada no

campo de perícia criminal, ajudando a desvendar o conteúdo de dados protegidos encontrados em investigações, sendo uma importante ferramenta na coleta de evidências (Pellegrini 2005). Outra aplicação da esteganálise é o de exploração de informações militares inimigas (Phister 2004). Ainda, as áreas de aplicação da esteganografia coincidem com as áreas de aplicação da esteganálise. Portanto, se a esteganografia é disponível para uso pessoal, admite-se que a esteganálise também pode ser utilizada por motivos pessoais (tais como a curiosidade).

1.4. Estrutura da Dissertação

Além da parte introdutória que foi apresentada neste Capítulo, o presente texto se organiza da seguinte forma:

- Capítulo 2 – discorre sobre as áreas que abrangem as técnicas abordadas na solução do problema; tais técnicas envolvem esteganálise de imagens digitais, sistemas multi-agentes, heurística de sociedades de abelhas, segmentação e reconhecimento de imagens, além de diferentes técnicas de aprendizado de máquina; em seguida são apresentados alguns trabalhos relacionados;
- Capítulo 3 – descreve a arquitetura do sistema multi-agentes desenvolvido, segundo as visões de caso de uso, de projeto, de processo, de implementação e de implantação; e em seguida descreve cada um dos agentes, de forma individual;
- Capítulo 4 – apresenta experimentos e resultados preliminares visando a escolha do conjunto de dados estatísticos otimizado; em seguida apresenta experimentos realizados no sistema como um todo, bem como seus resultados, ilustrando como o sistema evolui através da exibição de iterações realizadas no experimento;
- Capítulo 5 – discute sobre os experimentos e resultados do trabalho, apresentando considerações finais e perspectivas.

Capítulo 2.

Estado da Arte

"Em certa medida, toda arte é uma abstração." Henry Moore

O problema tratado neste trabalho envolve diferentes áreas de conhecimento, tais como criptologia, aprendizado de máquina, sistemas multiagentes, complexidade de algoritmos, insetos sociais, segmentação e reconhecimento de imagens. Neste Capítulo, mostramos os vários aspectos envolvidos na solução do problema abordado.

2.1 Criptologia

Criptologia é uma das sub-áreas do campo da segurança de informação que engloba criptografia, criptoanálise, esteganografia e esteganálise. Este trabalho é focado apenas em esteganálise. Mas, para compreender a esteganálise precisamos antes compreender a esteganografia, que é o que discutiremos primeiro.

2.1.1 Breve Histórico da Esteganografia

Esteganografia, palavra originária do grego, significa escrita coberta. Generalizando, consiste em ocultar informação em um meio que não levante suspeita da existência da informação. Define-se esteganografia digital como a arte de se ocultar dados, de forma insuspeita, em meio a outros dados digitais (Artz 2001).

Uma das técnicas de esteganografia usada pelos antigos gregos era a de cobrir as mensagens escritas em tábuas com cera escrevendo-se mensagens sem importância do outro lado, podendo então transporta-las sem levantar suspeita. Outra técnica utilizada por eles era a de raspar o cabelo do mensageiro, tatuar a mensagem secreta no couro cabeludo com alguma tinta especial, e quando o cabelo crescesse o suficiente para cobrir a mensagem ele poderia ser enviado para o destinatário da mensagem, carregando uma mensagem qualquer para despistar. Também se usava o transporte de mensagens escondidas em animais ou objetos. Aproximadamente na mesma época, essas técnicas rudimentares de esteganografia também eram utilizadas em outros locais, como Egito,

Mesopotâmia, China e Índia, sendo no Egito onde mais provavelmente a esteganografia começou a ser utilizada (Kipper 2004).

Até a Renascença usava-se apenas criptografia elementar, por substituição, quando surge o primeiro livro impresso sobre Esteganografia: Steganographia de Johannes Trithemius (1462-1516) (Figura 2.1). O trabalho foi escrito em manuscritos por volta de 1500 e posteriormente impresso, cujo conteúdo aparentemente esotérico contém, em parte esteganografadas e em parte cifradas, descrições de técnicas de criptografia e esteganografia, as mensagens esteganografadas no livro possuem frases de humor sádico.

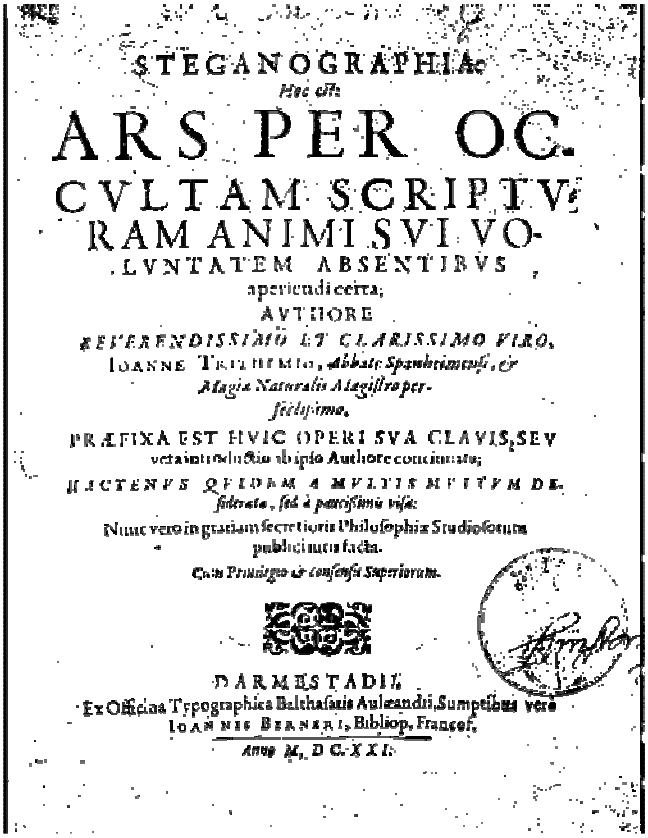
STEGANOGRAPHIAE
Hoc eſt:
ARS PER OC-
CVLTAM SCRIPTV-
RAM ANIMI SVI VO-
LVNTATEM ABSENTIBVS
aperiendi certa;
AVTHORE
REVERENDISSIMO ET CLARISSIMO VIRO,
IOANNE TRITHEMIO, Abbate Spanhemenſi, &
Magiæ Naturalis Magiſtro per-
ſcitiſſimo.
PRÆFIXA EST HVIC OPERI SVA CLAVIS, SEV
ver a intellectio ab ipſo Authore concinnata;
HACTENVS QVIDAM A MVLTIS MVLTVM DE-
ſiderata, ſed à paucißimis viſa:
Nunc vero in gratiam ſecretioris Philoſophiæ Studioſorum
publici iuris facta.
Cum Priuilegio & conſenſu Superiorum.
DARMESTADII,
Ex Officina Typographica Balthaſaris Auleandri, Sumptibus vero
IOANNIS BERNARI, Bibliop. Francof.
Anno M. DC. XXI.

Figura 2.1 – Capa do Livro "Steganographia" de Trithemius.

Outro nome importante na Renascença foi Gaspari Schotti (1608-1666), que escreveu Schola Steganographica, um volume de 400 páginas que expandia as técnicas de Trithemius. Era composto por 40 tabelas em diferentes idiomas com o alfabeto da época, que substituía cada letra de um texto que se queria ocultar por uma palavra, gerando uma mensagem esteganográfica que parecia uma oração (método de geração de cobertura).

Com o surgimento dos meios digitais de comunicação, logo as técnicas de esteganografia existentes foram adaptadas para este novo meio, e diversos novos tipos de técnicas de esteganografia exclusivamente digital surgiram nas últimas décadas.

2.1.2 Esteganografia Digital

O objetivo principal da esteganografia digital é esconder dados em uma comunicação de forma que interceptores não suspeitem de sua existência, nem do método empregado para ocultá-los.

Quando duas partes, *A* e *B*, desejam trocar uma mensagem secreta *m*, eles usam uma mensagem de cobertura *c* (também chamada de objeto de cobertura, envelope, mula ou simplesmente cobertura), aplicando sobre *c* uma técnica de esteganografia, que pode ou não usar uma chave *k*, obtendo assim uma mensagem esteganográfica *c*' (ou objeto esteganográfico). Este objeto é aparentemente indistinguível de c é então usado na comunicação entre eles.

Um objeto de cobertura geralmente pode ser um texto, imagem, áudio ou vídeo, sendo as imagens digitais o meio mais comumente utilizado nos dias atuais. Para um sistema perfeito, um objeto de cobertura normal não pode ser distinguido de um objeto esteganográfico.

A e *B* devem, logicamente, considerar que durante a comunicação onde eles transmitem o objeto esteganográfico, pode haver um interceptor ou atacante *I*.

Existem várias técnicas de esteganografia, sendo elas geralmente classificadas como:

- *Sistema de substituição* – partes redundantes ou menos importantes da mídia de cobertura são substituídas pelos dados que se deseja ocultar; um exemplo de dados menos importantes são os bits menos significativos de uma imagem de mapa de bits;
- *Técnica no domínio da transformada* – insere os dados secretos no domínio da transformada do sinal (domínio da freqüência);
- *Técnica de espalhamento de espectro* – espalha-se o espectro da informação que se deseja ocultar;

- *Métodos estatísticos* – esteganografa os dados que se deseja ocultar através da manipulação de dados estatísticos da cobertura, utilizando testes de hipóteses no processo inverso;
- *Técnicas de distorção* – gera distorções numa cobertura para esteganografar os dados, compara a cobertura original com a modificada para extraí-los (desesteganografá-los);
- *Métodos de geração da cobertura* – cria uma cobertura a partir dos dados que se deseja ocultar.

Dentre essas técnicas, podemos destacar as que operam no domínio da transformada e as que trabalham com métodos estatísticos, pois recentemente originaram ferramentas esteganográficas que tentam passar despercebidas por algumas das mais modernas técnicas de esteganálise existentes (Tzschoppe 2003), criando um ciclo vicioso ou virtuoso (dependendo do ponto de vista) com os esteganalistas.

Existem basicamente três tipos de protocolos de comunicação através de esteganografia (Katzenbeisser 2000): *esteganografia pura, esteganografia de chave secreta e esteganografia de chave pública*. Cada um desses protocolos indica de que forma uma mensagem *m* é oculta em uma cobertura *c*, utilizando-se de uma função de esteganografia *E*, resultando em *c'* (a cobertura com a mensagem oculta em si); e também indica a inversa E^{-1} que faz o processo de "desesteganografia" de *c'* resultando novamente em *c* e *m*.

As técnicas que utilizam o protocolo de esteganografia pura são as mais rudimentares, baseiam-se apenas no princípio da "segurança por obscuridade". Ou seja, são técnicas empregadas em situações onde o simples fato de ocultar a mensagem atrás de qualquer cobertura já é o suficiente para o nível de proteção que se deseja obter.

Em situações onde se requer maior segurança elas não devem ser empregadas, pois, para extrair os dados ocultos de uma cobertura, basta testar as funções inversas das esteganografias conhecidas sobre os dados que se está investigando e fatalmente os dados ocultos serão revelados.

As técnicas que se utilizam de chaves secretas são um pouco mais seguras. Para se ocultar ou extrair os dados ocultos, deve-se aplicar uma chave sobre a função de esteganografia ou sua inversa. O nível de segurança dessas técnicas depende do grau de segurança da chave utilizada. E as técnicas que usam chaves públicas são ainda mais

seguras, pois fazem uso de esquemas de chaves assimétricas (públicas e privadas) que são necessárias para, respectivamente, esteganografar e para sua inversa. Quando *A* deseja enviar uma mensagem para *B* utilizando esteganografia de chave pública, *A* usa a chave pública de *B* para esteganografar a mensagem e a envia com a cobertura para B; que por sua vez só pode extrair a mensagem se utilizar sua chave privada.

O funcionamento do esquema de chaves assimétricas nesse tipo de técnica de esteganografia é semelhante ao esquema de chaves assimétricas usado nos algoritmos de criptografia por chaves assimétricas como o RSA (Jonsson 2003) e o PGP (Zimmermann 1995). De fato, este tipo de esteganografia também faz uso de algoritmos de criptografia sobre os dados protegidos antes de ocultá-los na cobertura – o que fornece mais segurança para a comunicação.

Dentre as aplicações de esteganografia digital podemos citar:

- Assinatura digital e marca d'água – usadas principalmente pelas indústrias de mídias, que esteganografam informações dentro de arquivos de mídia como mp3 ou fotos para posteriormente rastrear de onde partiu a distribuição de cópias "piratas" ou informais de seus dados;
- Segurança de informação – usada por governos, indústrias, e organizações não governamentais, paramilitares, ou por indivíduos para segurança de informações pessoais.

2.1.3 Esteganálise

Enquanto a esteganografia visa ocultar a existência das informações que se deseja proteger, a esteganálise visa atacar essa forma de comunicação. Esse ataque pode ser de diversos tipos: detectar a existência de informações ocultas, adulterar ou forjar a mensagem oculta, interromper a comunicação, extrair os dados ocultos. Este trabalho, a princípio, se concentra nos ataques de detecção de dados ocultos. Podemos condensar as classificações dos métodos de esteganálise por detecção, a partir das classificações de Fridrich (2002), Katzenbeisser e Chandramouli (2004) como:

- *Ataque visual* – os métodos de esteganografia em imagens digitais mais elementares, como os sistemas substituição de bits podem causar distorções

visíveis que revelam o conteúdo escondido, caso as imagens sejam analisadas e comparadas plano a plano (de bits);

- *Análise estatística* – analisam dados estatísticos do arquivo ou de seu histograma para verificar se existem alterações incomuns, e que possam indicar a presença de dados ocultos; ainda pode ser classificada como *análise estatística pura, ou análise estatística por aprendizado de máquina* (que é o tipo de técnica que usamos neste trabalho);

- *Detecção de assinatura* - qualquer degradação causada por alguns métodos de esteganografia nos dados de cobertura pode ser lida como uma "assinatura" desses métodos; os métodos de esteganálise por detecção de assinatura costumam varrer os arquivos suspeitos em busca de "assinaturas" nos ruídos dos dados que possam revelar se algum método de esteganografia foi utilizado e até qual método específico de esteganografia foi utilizado.

Além da técnica empregada, esses métodos podem ser classificados ainda, tanto no tocante aos algoritmos de esteganografia que visam detectar, quanto pelos tipos de dados lidos, como:

- *Específica* – métodos de esteganálise que são orientados para atacar uma ou um grupo de técnicas de esteganografia em específico, como por exemplo:

  a) *Análise de histograma*, para detectar o uso de algoritmos de esteganografia que usam de quantização nos histogramas para ocultar os dados;

  b) *Verificação de paletas*, para detectar anomalias na ordenação de paleta que alguns algoritmos de esteganografia causam;

  c) *Verificação de pares de pixels*, para detectar ocorrências anormais de pares de pixel de mesmo valor que pode ocorrer com o uso de técnicas de esteganografia no bit menos significativo;

- *Universal* – ou esteganálise cega, métodos de esteganálise que são projetados para detectar a existência ou não de dados escondidos, independente da técnica empregada para ocultá-los.

Wang (2004) faz uma lista de várias técnicas de esteganálise frente à métodos de esteganografia conhecidos. Destacamos os métodos universais baseados em estatísticas de alto nível das imagens. Essas estatísticas são a média, a variância, a assimetria e a

curtose, apresentadas na seção 2.4. Esses métodos normalmente treinam um mecanismo de aprendizado de máquina a partir das estatísticas citadas para resolução do problema de classificação binária. Ou seja, se o dado analisado possui ou não possui uma mensagem esteganografada.

No presente trabalho, além de usar métodos de esteganálise universais baseados em estatísticas de alto nível para a classificação binária citada, também precisamos descobrir qual ferramenta de esteganografia provavelmente foi usada para ocultar e proteger os dados para tentar extrair esses dados, o que pode ser obtido também através de classificação. Mais detalhes sobre essas técnicas de classificação podem ser vistos na seção a seguir.

2.2 Inteligência Artificial

Na Seção 2.1. vimos que algumas técnicas de esteganálise utilizam técnicas de aprendizado de máquina para detectar a presença de dados ocultos em dados "comuns". Além de aprendizado de máquina, outro tópico de inteligência artificial que é abrangido por este trabalho são os sistemas multiagentes, que utilizamos para aumentar a eficiência dos algoritmos de AM e para solucionar o problema de forma distribuída.

2.2.1 Sistemas Multiagentes

Para compreender o que são Sistemas Multiagentes, primeiro precisamos compreender o que são agentes. Existem diversas definições de agentes de software. Bradshaw (1997) por exemplo, apresenta duas definições de agente: "Agente é o que um agente faz", e "uma entidade de software que trabalha continua e autonomamente em um ambiente em particular, geralmente habitado por outros agentes e processos". A definição mais conhecida de agentes é a de Russel (1995), que diz que "agentes são entidades de software autônomas que atuam em um ambiente, capazes de decidir que ações tomar para atingir seus objetivos".

Franklin (1996) classifica os agentes de acordo com as propriedades que eles apresentam, podendo um agente ser classificado por mais das propriedades a seguir:

- *Reatividade* – os agentes são ditos reativos, quando suas ações são meras reações ou reflexos frente às mudanças no ambiente;

- *Autonomia* – os agentes autônomos são os que controlam as próprias ações que irão tomar;
- *Pró-atividade* – agentes pró-ativos não agem somente em resposta ao ambiente;
- *Continuidade temporal* – agentes que são executados em processos contínuos;
- *Comunicação* – agentes comunicativos comunicam-se com outros agentes e podem se comunicar com pessoas;
- *Aprendizado* – aprendizes são os agentes que mudam seu comportamento baseados em suas experiências passadas;
- *Mobilidade* – os agentes móveis são capazes de se transportar de uma máquina para outra;
- *Flexibilidade* – as ações dos agentes flexíveis não seguem um *script*;
- *Personalidade* – apresentam personalidade e estados emocionais em suas atitudes ou mensagens.

Sistemas Multiagentes, ou SMAs, são sistemas que apresentam um conjunto de agentes de software que interagem entre si (através de comunicação) e com o ambiente; agentes possuem áreas de influência sobre o ambiente; essas áreas de influência podem ou não coincidir (Wooldridge 2001).

Rezende (2003b) apresenta uma classificação para Sistemas Multiagentes, de acordo com 5 eixos:

- *Perspectiva* – um SMA pode ter uma perspectiva científica de simulação ou resolução social, cujas aplicações são o estudo das interações sociais, ou resolução de problemas de forma cooperativa e distribuída;
- *Abertura* – indica se um SMA é aberto ou não à mudanças dinâmicas do número de agentes que o compõe;
- *Granularidade* – SMAs com poucos agentes são ditos SMAs de baixa granularidade, e SMAs com muitos (milhares) agentes são ditos SMAs de alta granularidade;
- *Composição* – indica se um SMA possui agentes homogêneos ou heterogêneos (de funcionalidade diferente);

- *Interação* – indica os tipos de interação social entre os agentes, que basicamente são:
    - *Neutralismo* – nenhuma interação entre os agentes;
    - *Competição* – quando agentes competem pelos mesmos recursos, fazendo com que um agente atinja negativamente o desempenho do outro ao utilizar tais recursos;
    - *Amensalismo* – acontece quando um agente afeta os recursos ou o funcionamento de outro por uma ação não proposital;
    - *Parasitismo* – acontece quando um agente depende de outro, agindo negativamente e de forma intencional sobre este, mas sem destruí-lo;
    - *Predação* – um dos agentes é elimidado ou neutralizado pelo outro;
    - *Comensalismo* – quando a interação entre os agentes beneficia um dos agentes e não causa ônus no outro;
    - *Proto-cooperação* – ambos os agentes se beneficiam com a interação, mas cada um pode atingir os mesmos benefícios gastando um pouco mais de tempo ou recursos;
    - *Simbiose ou mutualismo* **–** ambos os agentes se beneficiam com a interação, e nenhum pode atingir os mesmos benefícios isolado.

Nesse tipo de sistema, os agentes podem interagir muitas vezes através de negociação, ou outras formas de coordenação ou cooperação. A negociação pode existir entre dois ou mais agentes; e dentre seus mecanismos, podemos citar (Macedo 2001):

- *Leilão* – existem diversas versões de negociação por leilão, são negociações formuladas de acordo com os diferentes tipos de leilões existentes (secreto, público, maior lance, menor lance, etc.), onde os agentes que querem disputar por um mesmo recurso fazem suas "ofertas" dizendo, por exemplo, o quanto aquele recurso é importante para ou oferecendo outros recursos em troca;
- *Argumentação* – um agente tenta influenciar as intenções do outro a partir da exposição de argumentos, numa negociação por argumentação podem haver, além das propostas, críticas e contra-propostas;

- *Teoria dos jogos* – dessa teoria citamos o *Protocolo de Concessão Monotônica* (PCM), onde os participantes negociam interativamente até encontrarem a solução de menor custo para ambos.

Um outro protocolo de negociação conhecido são as **redes de contrato** (Paurobally 2002), cujo funcionamento é semelhante ao das licitações do governo brasileiro. Um agente gerente solicita propostas de negócio para os demais agentes, que respondem com uma mensagem *proposta*, *não compreendido,* ou *negado*. Dentre os agentes que enviarem uma proposta, alguns serão selecionados (contratados) e receberão uma mensagem *proposta-aceita* e os demais *proposta-rejeitada*. Os que foram contratados deverão ao final da execução da tarefa enviar uma mensagem informando que a tarefa foi concluída.

Existem diversas formas de coordenação de agentes, podendo uma sociedade de agentes ser composta somente por agentes homogêneos ou heterogêneos dividindo as tarefas que precisam ser executadas. O problema da coordenação consiste em como gerenciar a interdependência de tarefas e recursos entre agentes. Wooldridge (2001) classifica os modelos de coordenação em:

- *Planejamento global parcial* – agentes cooperativos trocam informação para atingir os objetivos comuns sobre o problema; primeiro cada agente decide suas próprias metas e gera seus planos locais para atingi-las; depois os agentes trocam informações pra descobrir onde seus planos e objetivos interagem com os dos outros; por último os agentes alteram seus planos locais para coordenar melhor suas atividades com as relacionadas;

- *Intenções conjuntas* – nesse modelo de coordenação os agentes devem se comprometer a atingir os objetivos juntos, a partir de compromissos individuais das tarefas específicas que eles foram designados; assim as intenções de cada agente em uma atividade coordenada possuem a carga da responsabilidade desse comprometimento;

- *Modelagem mútua* – nesse modelo cada agente possui um modelo interno das crenças, intenções e etc. dos demais agentes; assim cada agente coordena suas ações a partir das predições feitas por esse modelo;

- *Normas e regras sociais* – esse modelo baseia-se na definição de convenções ou regras sociais que delimitam que atitudes um agente pode ou não tomar

num SMA; as regras podem ser projetadas previamente, ou podem emergir em tempo de execução dentro do sistema.

Para realizar a coordenação, os agentes também precisam se comunicar. Uma das formas de comunicação entre agentes é o uso de *blackboards* ou quadros-negros, que têm esse nome por metáfora aos quadros-negos usados em salas de aula. Eles consistem basicamente em repositórios de dados usados para gerenciar as interações entre grupos de agentes em um SMA (Lander 1997). Em outras palavras, um quadro-negro apresenta uma interface onde cada agente pode deixar mensagens para ser lidas por um ou por todos os demais agentes. Um SMA que utiliza este tipo de ferramenta de comunicação entre os agentes, deve possuir um agente especial, responsável por coordenar as atividades dos demais agentes, enviando a eles mensagens de controle quando necessário ou notificando aos agentes eventos que lhes interessem. Esse agente é chamado de *control shell*.

Nosso método de coordenação de agentes baseia-se principalmente no *modelo de intenções conjuntas*, mas um nível de abstração diferente também pode ser classificado como um método que segue *normas e regras sociais*. A negociação é feita baseada em dois protocolos. Na ativação da negociação, usa-se o *Protocolo de Redes de Contrato*. Após definidos quais agentes serão incumbidos da tarefa, tais contratados fazem uso do *Protocolo de Concessão Monotônica* para chegar a um consenso em seu resultado final.

2.2.2 Aprendizado de Máquina

"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" (Rezende 2003a).

O aprendizado (computacionalmente falando) é a aquisição de conhecimento através da *dedução* do conhecimento existente ou da *indução* de novo conhecimento (Sison 1998). Os algoritmos de *aprendizado por dedução* inferem novas regras de conhecimento a partir das regras de conhecimento que já possuem, geralmente com o propósito de melhorar seu desempenho em termos de tempo. Isto é conseguido ao se reduzir o número de regras (via inferências de lógica de 1ª ordem) usadas para se chegar

a uma conclusão, ou ao se usar heurísticas que façam uma busca otimizada nessas regras.

Os algoritmos de *aprendizado indutivo*, também chamados de algoritmos de *aprendizado de máquina*, são os algoritmos que produzem hipóteses e generalizam problemas a partir de dados de exemplos; e se dividem basicamente entre algoritmos de aprendizado supervisionado, de aprendizado não-supervisionado, e de aprendizado baseado em recompensas. Conjuntos de dados são submetidos para *treinar* estes algoritmos, contendo exemplos de classificação ou regressão de problemas, para que estes algoritmos possam *generalizar* ou *aproximar* a função que resolve tais problemas. Cada *exemplo* de treinamento contém diversos *atributos* ou parâmetros relacionados ao problema que abordam.

Nos algoritmos de *aprendizado supervisionado*, os exemplos de treinamento também indicam a qual classe cada *exemplo* ou *instância* pertence. Após o treinamento, os algoritmos de aprendizado de máquina supervisionado são capazes de receber novos exemplos de classes desconhecidas, e então inferir a que classe esses novos exemplos pertencem.

Os algoritmos de *aprendizado não-supervisionado* geralmente são treinados para *agrupar* conjuntos de exemplos com propriedades semelhantes em classes; e quando são submetidos à novos dados, indicam a que classe (grupo) os novos exemplos devem pertencer.

No *aprendizado baseado em recompensas*, (Barrios-Aranibar 2005, Panait 2005) os algoritmos aprendem a solucionar um problema a partir de tentativas de se chegar a um resultado final que lhes é requisitado. Estes algoritmos se dividem entre algoritmos de *aprendizado por reforço,* que estimam pesos de estados ou de estado/ação indicando quais estados ou estados/ações são melhores para a resolução de um problema, e de *busca estocástica*, que aprendem comportamento de funções diretamente sem apelar para ponderação (simplesmente avaliando o ganho de seu resultado).

Quando os problemas abordados pelos algoritmos de aprendizado de máquina envolvem uma quantidade definida de classes, são chamados de problemas de *classificação*, e quando o conjunto de classes é contínuo ou indefinido, os problemas são conhecidos como problemas de *regressão* numérica. E um algoritmo de aprendizado de máquina que faz classificação também é chamado de classificador.

Dentre os paradigmas de aprendizado de máquina por indução, quer seja os supervisionados, não-supervisionados, ou baseados em recompensa, temos (Sanches 2004):

- *paradigma simbólico* – constrói uma representação simbólica da resolução do problema através da análise de exemplos, o métodos de aprendizado de máquina desse paradigma são as árvores de decisão, e as redes semânticas;
- *paradigma estatístico* – são os métodos de classificação que buscam analisar estatísticas para encontrar um modelo estatístico aproximado ou generalizado do problema; dentre os métodos mais conhecidos desse paradigma estão o aprendizado Bayesiano e os *Support Vector Machines* (SVM);
- *paradigma baseado em exemplos* – classifica uma instância (exemplo) através da comparação desta com as instâncias já classificadas, retornando a classe da instância classificada que se aproxime mais dela; o métodos mais conhecido desse paradigma é o K-nn (K nearest neighbours) que retorna a classe que mais aparecer nos K vizinhos mais próximos de um grupo para um exemplo consultado;
- *paradigma conexionista* – baseia-se na metáfora biológica das conexões neurais do sistema nervoso, busca treinar uma rede de neurônios com exemplos de tal forma que seus pesos em suas conexões se ajustem para resolver o problema de classificação;
- *paradigma genético ou evolutivo* – também baseia-se em uma metáfora biológica, a da evolução genética; os algoritmos genéticos também podem ser classificados como algoritmos de aprendizado baseado em recompensa ou por reforço.

Os paradigmas simbólico e conexionista são os mais comumente usados para construção de métodos de esteganálise, conforme veremos no final deste capítulo.

Em alguns casos os algoritmos de AM podem se ajustar demais ao conjunto de dados de treinamento, situação esta chamada de *overfitting*, que pode aumentar as taxas de acerto do algoritmo durante o treinamento, mas em testes com exemplos não apresentados no treinamento revela-se que as taxas reais de acerto são bem inferiores.

Cada método de aprendizado, depois de treinado, apresenta uma taxa de erro ou acerto. E normalmente se deseja melhorar as taxas de acerto e evitar demais problemas que possam surgir em decorrência do treinamento; e uma das estratégias para isso são os *ensembles* ou *agrupamentos de classificadores* que são algoritmos de aprendizado que treinam conjuntos de classificadores para classificar novas instâncias através de votação que em alguns casos é ponderada (Dietterich 2000). Eles podem ser usados para evitar situações indesejadas, como por exemplo, quando os classificadores se prendem em mínimos ou máximos locais. Duas das técnicas (ou algoritmos) de agrupamento mais conhecidas são o *bagging* e o *boosting*.

Bagging ou *Bootstrap Aggregating* é um algoritmo que consiste em usar um conjunto de treinamento de tamanho *m* para gerar diversos subconjuntos (com reamostragem) também de tamanho *m*, cada um desses subconjuntos é usado para treinar os algoritmos de AM que compõe a banca de votação no processo de classificação (o mesmo tipo de algoritmo é usado com um dos diferentes subconjuntos de treinamento para cada um dos classificadores). O *Bagging* é usado para métodos de aprendizado que são mais instáveis (variam muito com pequenas mudanças no conjunto de treinamento) como as árvores de decisão e as redes neurais, tornando-os mais estáveis e evitando também a ocorrência de *overfitting*.

Boosting é um algoritmo de agrupamento iterativo: a cada iteração um classificador é treinado para um conjunto de dados**,** o classificador recebe um peso de acordo com o seu desempenho, e cada um dos dados de treinamento também recebe um peso de acordo com a dificuldade em classificá-los; este classificador é adicionado ao classificador da iteração seguinte, que é treinado pelo conjunto de treinamento ponderado na iteração anterior; e assim sucessivamente, resultando em um classificador final que é o somatório ponderado dos classificadores usados em cada iteração.

2.2.3 Algoritmos de Aprendizado de Máquina

A seguir descrevemos alguns dos algoritmos de aprendizado de máquina mais conhecidos, representando cada um dos paradigmas de aprendizado de máquina por indução apresentados.

As ***árvores de decisão*** (Quinlan 1986) são algoritmos usados para problemas de classificação ou regressão, que representam regras de indução através de grafos em

formatos de árvores. Cada nó da árvore representa uma das sub-regras que compõe a regra empregada para se chegar a uma das folhas, que contém a classe que aquela regra produz.

Para construir estes grafos, as árvores são treinadas por conjuntos exemplos, que gradualmente são separados a cada nó pelos valores de um dos atributos desses exemplos (chamado de *operador*), até se chegar às folhas, cujos valores correspondem à classe em que todos os exemplos de cada subgrupo representa.

Em alguns casos, essas regras tornam-se muito longas, o que pode prejudicar a generalização, causando *especialização*; para evitar esse problema, as árvores podem sofrer *poda*, impedindo que seus ramos ultrapassem certo limite de tamanho durante o treinamento. Na poda, a classe que cada folha possui corresponde à classe em que a maioria dos exemplos de cada subgrupo representa.

Para se classificar um exemplo, ele deve ser submetido à árvore de decisão treinada, e então percorre diversos nós dessa árvore, até chegar em uma folha. Uma das vantagens das árvores de decisão, é que o caminho percorrido na árvore até se chegar à folha pode ser usado para explicar como a decisão (de que classe aquele exemplo pertence) foi tomada. As regras de uma árvore de decisão também podem ser visualizadas graficamente, o que também facilita a compreensão de seu funcionamento.

SVM, ou ***support vector machines*** (Burges 1998), são algoritmos de aprendizado de máquina supervisionados indutivos do paradigma estatístico, geralmente usados para problema de classificação, mas também podem ser usados para regressão.

Seu funcionamento baseia-se em separar os exemplos de um problema através de hiperplanos (Figura 2.2), com as maiores margens possíveis de distância entre si (hiperplanos ótimos). As SVM mais simples são classificadores binários, ou seja, separam os exemplos entre os que pertencem a uma classe A e os que pertencem a uma classe B.

Cada instância de treinamento que compõe um dos vetores que separa os hiperplanos é chamada de *support vector*. Se os dados de treinamento forem linearmente separáveis, não haverá pontos entre os hiperplanos, para os demais casos foram criadas as SVMs não-lineares. Após treinada, a SVM pode receber novas instâncias e classifica-las de acordo com o hiperplano a que essa instância se encaixar.

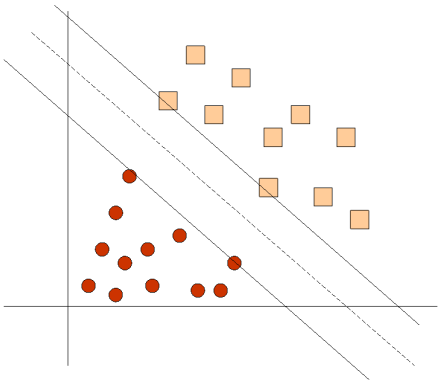

Figura 2.2 – Instâncias separadas em classes linearmente separáveis, através de hiperplanos em um SVM.

K-nn (K-nearest neighbour), (Zhang 2006) é um algorítmo de aprendizado de máquina supervisionado baseado em exemplos, usado para classificação ou regressão. Os exemplos de treinamento formam uma espécie de tabela de grupos que é acessada toda vez que um dado for classificado por este algoritmo.

A base do funcionamento do K-nn é a função de distância, que indica o quanto um exemplo é diferente do outro. Durante a classificação, a classe (grupo) da maioria dos K exemplos de menor distância à instância que está sendo classificada, é indicada como a classe dessa instância.

Dentre as redes neurais mais conhecidas estão as redes ***MLP*** (***Multi-Layer Perceptron)*** ou Perceptron Multi-camadas, (Patiño-Escarcina 2004), que utilizam pelo menos uma camada de neurônios (camada escondida) entre a camada de neurônios que recebe os dados de entrada (camada de entrada) e a camada de neurônios que retorna os valores de saída (camada de saída).

Os *neurônios* são as estruturas básicas de processamento de uma rede neural, e no modelo ***Perceptron*** cada neurônio possui diversas conexões de entrada e apenas uma conexão de saída. As conexões de entrada que cada neurônio possui são ponderadas, e seu peso é ajustado durante a fase de treinamento. Cada neurônio possui uma *função de ativação*, que calcula o valor que enviará à sua conexão de saída a partir dos valores e dos pesos de suas conexões de entrada.

Os neurônios da camada de entrada possuem suas conexões de entrada diretamente ligadas a cada um dos valores dos atributos de uma instância do problema.

As saídas dos neurônios dessa camada conectam-se as entradas dos neurônios da primeira camada escondida, cujas saídas por sua vez conectam-se as entradas da próxima camada de neurônios – que pode ser uma nova camada escondida (se existir), que pode conectar-se a outra camada escondida (se existir) e assim por diante – até que as saídas dos neurônios da última camada escondida se conectem às entradas da camada de saída (Figura 2.3). Cada neurônio da camada de saída mapeia se uma dada entrada pertence ou não a uma classe.

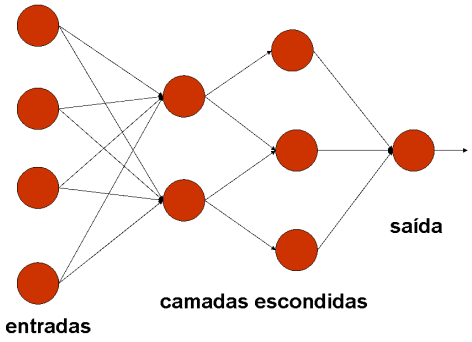

Figura 2.3 – Exemplo de Rede Neural MLP com duas camadas escondidas.

Essas redes são do tipo *feedfoward*, ou seja, as informações fluem da camada de entrada até a camada de saída de forma acíclica. Entretanto, os erros de classificação durante o treinamento são propagados através de retro-alimentação.

Os ***algoritmos genéticos*** consistem em realizar cruzamentos e mutações num conjunto de classificadores para solucionar um problema, durante N interações (ou gerações), os classificadores com melhor desempenho em cada geração prevalecem e os classificadores das gerações seguintes são variações destes; os genes são na verdade parâmetros dos classificadores, que por sua vez são de qualquer um dos outros paradigmas, só que no lugar dos classificadores adaptarem estes parâmetros através de treinamento com instâncias, eles mudam esses valores através da evolução.

2.3. Complexidade de Algoritmos e Colônias de Abelhas

2.3.1 Computação

Heurísticas são soluções mais simples que a solução completa de um problema, usadas para encontrar uma solução aproximada desse problema em um curto espaço de tempo.

Em computação é comum o uso de heurísticas para resolver problemas cuja complexidade não permitiria sua resolução de forma linear em tempo computacional hábil.

Dentre as heurísticas mais comumente utilizadas estão os algoritmos genéticos, algoritmos meméticos, busca tabu, anilhamento simulado, e colônias de insetos sociais como formigas e abelhas (Siqueira, 2005). Os algoritmos que utilizam as metáforas das sociedades de insetos buscam simular algum aspecto de seu comportamento social para solucionar problemas como o de encontrar o menor caminho num grafo.

Existem algumas técnicas para se melhorar o desempenho desses algoritmos, como o uso empírico de experimentos para obter uma melhor configuração dos mesmos (Hussain 2005). Um fator crucial desses experimentos é encontrar um conjunto de dados de treinamento que *represente* o problema de forma suficiente para encontrar-se uma solução boa ou ótima.

2.3.2 Biologia

Insetos sociais podem ser monogínicos ou poligínicos, ou seja, podem existir sociedades que apresentam apenas uma rainha ao mesmo tempo em que podem existir sociedades que possuem várias rainhas.

Aponte (2003) apresenta em sua tese de doutorado um estudo sobre o comportamento social das abelhas Meliponine Bicolor (Fig. 2.4). Essa espécie apresenta algumas colméias monogínicas e outras poligínicas. Sua estrutura social evolui da seguinte forma, de acordo com a própria autora:

"O esforço conjunto de duas ou mais rainhas produz um aumento significativo na taxa geral de postura, mas com uma contribuição menor por cada rainha. Várias rainhas atraem um número maior de operárias que, estimuladas pela necessidade de ovipositar das rainhas, aprovisionam com rapidez, até em células sem colar. Este último feito obriga ao investimento de maiores esforços para o fechamento de células, representados por um número maior de operculadoras."

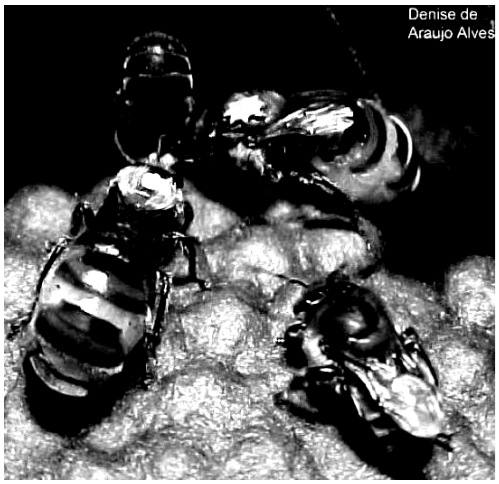

Figura 2.4 – Grupo de rainhas de M. Bicolor em processo de postura, ambas auxiliadas por uma operária em comum.

Em outras palavras, as rainhas convivem pacificamente entre si, e coordenam as tarefas das operárias sem fazer distinção de sua origem, ocasionalmente a demanda de algumas tarefas especializadas acarreta no surgimento de operárias especializadas para atendê-la; e o aumento das colméias também pode demandar o surgimento de novas rainhas para auxiliar o trabalho das já existentes, gerando uma retro-alimentação positiva.

2.4. Processamento de Imagens

Dentre as técnicas de processamento de imagem existentes, discorremos aqui sobre segmentação, por interessar diretamente ao assunto em pauta. Segmentação de imagens é o processo de particionar imagens em múltiplas regiões, ou conjuntos de *pixels*, de acordo com um critério dado. O objetivo da segmentação geralmente é localizar padrões ou objetos graficamente nas imagens. A esteganálise, em alguns casos, usa segmentação para detectar distorções visualmente imperceptíveis dentro desses padrões.

O processo de segmentação de imagens que utilizamos consiste em colher dados estatísticos sobre as camadas R, G, B, H, S e V referentes respectivamente às camadas de cores vermelha, verde e azul do modelo de cores RGB, e às camadas de intensidade,

saturação e brilho do modelo HSV. Do histograma das camadas citadas, são avaliadas sua média, variância, assimetria e curtose. Chegamos a esse conjunto de dados estatísticos a partir de experimentos realizados sobre os dados usados nos trabalhos de Tzschoppe (2003), Lyu (2006), Fridrich (2004), Miche (2006) e Wang(2007) descritos mais adiante em "trabalhos relacionados".

A média, como o próprio nome já diz, é o valor médio da distribuição normal analisada, no gráfico da distribuição normal, coincide com o ponto médio da curva normal e também é chamada de valor esperado.

Variância é a medida de dispersão dos valores encontrados na distribuição de seu valor esperado, ela mostra o quanto a curva se afasta do ponto médio.

Assimetria ou Obliquidade, segundo Hair (1998), é a medida da assimetria de uma distribuição; uma distribuição positivamente assimétrica tem relativamente poucos valores altos e é disposta mais à esquerda, uma distribuição negativamente assimétrica tem relativamente poucos valores baixos e é disposta mais à direita.

A curtose indica o quanto a curva de uma distribuição é achatada ou afinada quando comparada a uma distribuição normal. Valores positivos indicam distribuições relativamente afinadas, e valores negativos indicam distribuições relativamente achatadas.

Variância, assimetria e curtose são, respectivamente o segundo, terceiro e quarto momentos sobre a média (o primeiro momento é o valor 0). A equação 2.1 mostra a fórmula geral para encontrar o k-ésimo momento em relação a média, e também pode ser lida como $E[(X - E[X])^k]$ onde X é um variável aleatória, e E[X] é o seu valor esperado (a média da distribuição).

$$\mu_k = \int_{-\infty}^{+\infty} (x-\mu)^k f(x) dx \qquad \text{(Equação 2.1)}$$

2.5. Trabalhos Relacionados

A corrida esteganografistas X esteganalistas é abordada em detalhes por Wang (2004). Mas, para ilustrá-la, podemos citar uma abordagem de esteganografia que é apresentada por Tzschoppe (2003), onde os dados são ocultos num algoritmo baseado em estatísticas de alta ordem do arquivo de cobertura, visando burlar sua detecção pelas técnicas de esteganálise universal, que justamente tentam detectar a presença de dados esteganografados analisando esses dados estatísticos.

Por outro lado, podemos ver em Lyu (2006) uma abordagem de esteganálise universal baseada justamente em análise de dados estatísticos de primeira e alta ordem de imagens digitais, mostrando que seu algoritmo é capaz de detectar a presença de dados ocultos em imagens onde estas informações estatísticas apresentam distúrbios. Os atributos usados por Lyu foram a média, a variância, a assimetria e a curtose de imagens, para orientações vertical, diagonal e horizontal, formando 36m estatísticas para $\boldsymbol{m}$ escalas dos 3 canais de cores (RGB). As mesmas 36m estatísticas referentes às estatísticas de erro de um preditor linear; e 6(m+1)m estatísticas de fases de uma transformada gaussiana.

Um exemplo curioso dessa corrida também pode ser visto em Sung (2004), que apresenta uma técnica de esteganografia que utiliza gramáticas livres de contexto para gerar uma cobertura que guarda a mensagem previamente criptografada, criada com o objetivo de evitar os ataques de adulteração da mensagem oculta. A cobertura criada no trabalho citado é uma animação e os dados são ocultos nos gestos que os personagens da animação fazem, ao longo dos *frames*. Nesse caso, a adulteração dos gestos de uma animação não é uma mudança sutil o suficiente para este tipo de ataque ser bem sucedido, por isso é uma técnica robusta de esteganografia contra ataques de adulteração dos dados.

Um exemplo de abordagem de esteganálise de análise estatística específica é o trabalho apresentado por Fridrich (2004), onde é feita uma abordagem de esteganálise para imagens digitais que foram esteganografadas por algum algoritmo de substituição LSB. O objetivo desse algoritmo é não só o de detectar se há dados ocultos, mas também de determinar o tamanho desses dados.

Os atributos usados por Fridrich são: *média do histograma*; *histograma individual médio para 5 valores da transformada DCT; histograma dual médio para*

11 valores da transformada DCT; variância do histograma; 2 camadas de detecção de blocos; co-ocorrência de 00 01 e 11.

Miche (2006) apresenta uma metodologia de seleção de atributos para classificação de esteganálise. Em seu trabalho, ele faz experimentos sobre os 23 atributos usados por Fridrich e apresenta um subconjunto de 14 atributos que demonstrou o mesmo desempenho que o conjunto original. Os atributos selecionados por Miche sobre Fridrich foram: *histograma dual para 7 valores da transformada DCT; média do histograma; co-ocorrência de 01 e 11; histograma individual médio de 3 valores; e 2ª. camada de detecção de blocos*.

O trabalho apresentado por Wang (2007) discute outra forma de otimização dos atributos utilizados em classificação de esteganálise. Primeiro eles decompõem as imagens, depois usam como atributos para a classificação os diversos momentos estatísticos sobre estas imagens, e em seguida avaliam e escolhem os melhores atributos desse conjunto, que são os quatro primeiros momentos estatísticos de alta ordem (média, variância, assimetria e curtose) para detecção de erros, ou os 3 momentos estatísticos da função característica apenas para esteganálise de *wavelets.*

Realizamos experimentos com diversas combinações dos dados usados nos trabalhos citados na tentativa de achar um subconjunto ótimo. O subconjunto encontrado utiliza a média, variância, assimetria e curtose dos canais R, G, B, H, S e V. Os detalhes sobre esses experimentos podem ser vistos no Capítulo 4.

As análises estatísticas podem ser realizadas por fórmulas matemáticas, por comparação de pares de dados, ou podem ser realizadas por mecanismos de aprendizado de máquina. Podemos ver um exemplo de algoritmo de esteganálise, que se utiliza de redes neurais artificiais para detectar a presença ou não de dados ocultos em imagens digitais a partir de dados estatísticos dos coeficientes das transformadas (DFT, DCT e DWT) dessas imagens no trabalho de Shaohui (2003).

Rocha (2006) utilizou como métodos de aprendizado as árvores de decisão, análise de descriminante linear, e as máquinas de vetores de suporte (*suport vector machines*) para realizar esteganálise específica no canal dos bits menos significativos de imagens no formato JPG. Sua análise é feita através de uma técnica de análise estatística progressiva, desenvolvida pelo próprio autor. O método de agrupamento de classificadores empregado (*bagging*) aparentemente não apresentou melhoras

significativas ao resultado do sistema. Porém, pelas propriedades desse método de agrupamento, pode-se dizer que a utilização do método serviu para aumentar a robustez de seu sistema esteganalítico contra o *overfitting*.

Uma ferramenta de esteganálise interessante pode ser vista no trabalho de Gonzalez (2002), que apresenta uma ferramenta de buscas de dados esteganografados pela internet. Em sua ferramenta pode-se verificar se uma determinada imagem contém ou não dados esteganografados, e também pode-se monitorar (através de técnicas de *sniffing*) a rede em busca de fluxo de imagens contendo dados esteganografados.

Dentre os métodos de esteganálise estudados, a taxa de acerto chega a ser alta. Os mais bem sucedidos atingem taxas de acerto que variam de 70% a 90%, porém a natureza da criptologia requer que, para dominar o campo, se assuma a liderança na corrida tecnológica entre enteganografistas e esteganoanalistas. Os agrupamentos usando mecanismos de aprendizado de máquina estão começando a ser usados para aumentar a confiabilidade das ferramentas de esteganálise.

Um dos fatores importantes para um sistema esteganalítico é a flexibilidade, a capacidade de adaptação da ferramenta de esteganálise para novos métodos que possam surgir. Essa flexibilidade pode ser obtida construindo-se um sistema modularizado, de fácil manutenção. Outro fator importante é conseguir uma taxa de acertos alta, e isso pode ser conseguido através da combinação de diferentes métodos de aprendizado de máquina, quer sejam diferentes algoritmos, ou o mesmo algoritmo treinado por conjuntos de instancias diferentes.

Para isso, utilizamos nesse trabalho um sistema multiagentes, cuja sociedade é coordenada de acordo com a metáfora das abelhas poligínicas, citada anteriormente. Essa coordenação segue as normas sociais previamente embutidas nos agentes de hierarquia (rainhas ***x*** operárias). Os agentes que possuem os classificadores têm ainda a necessidade de negociar entre si para concordar em uma mesma classificação, nesse caso o esquema de negociação utilizado é o PCM.

A flexibilidade e a combinação de métodos de aprendizado pode ser obtida distribuindo os classificadores entre diversos agentes coordenados em um SMA. Este sistema multiagentes é organizado de tal forma que torna possível a adição de futuros agentes treinados para detectar os futuros métodos de esteganografia.

Capítulo 3.

Sistema de Agentes Poligínicos para Esteganálise de Imagens Digitais

"Os componentes da sociedade não são os seres humanos, mas as relações que existem entre eles" Arnold Toynbee

O sistema multiagentes desenvolvido e aqui descrito pode ser chamado de um sistema multi-agentes autônomos de software orientados a tarefas específicas, pela taxonomia de Franklin (1996). E de acordo com a taxonomia abordada por Rezende (2003), pode ser dito como um sistema de agentes heterogêneos aberto, com granularidade inicial baixa. A principal perspectiva do sistema é a de resolução social que visa resolver o problema de esteganálise de forma cooperativa e distribuída, porém também aborda um pouco a perspectiva de simulação social no tocante ao intuito de simular o comportamento social das abelhas poligínicas. Os padrões de interação presentes entre os agentes desse sistema são:

- Comensalismo – nas interações das operárias com as rainhas;
- Proto-cooperação – nas interações entre as agentes operárias.

Em geral, cada imagem é uma instância que é submetida ao sistema, o fluxo das instâncias no sistema é seguido de uma das formas abaixo.

1. O ambiente é criado e recebe um conjunto de dados de treinamento e um conjunto de dados para classificar (ou conjunto de teste).
2. Nascem algumas rainhas (a quantidade de rainhas é definida de acordo com a disponibilidade para alocação dos múltiplos processos no computador).
3. A rainha que não possuir uma instância para coordenar sua classificação verifica se o ambiente necessita da classificação de alguma das instância do conjunto de testes. Caso não haja mais instância que precise de classificação, a rainha morre. Caso haja instância para ser treinada, o

ambiente indica à rainha uma das instâncias do conjunto de testes que não estiver sido classificada e a rainha vai para o passo 4, a seguir.

4. A rainha faz uma chamada para verificar se existem operárias disponíveis para trabalhar a classificação da instância para ela. Caso uma quantidade satisfatória de operárias não responda à chamada, a rainha entra em processo de postura das operárias que, em seguida, nascem no ambiente. A quantidade de operárias é definida de acordo com a disponibilidade para alocação dos múltiplos processos no computador. A rainha repete o passo 4 até que uma quantidade satisfatória de operárias atenda a sua chamada e então vai para o passo 5, descrito a seguir.

5. As operárias classificam a instância atribuída. A rainha que coordena a classificação dessa entrada coordena a negociação das operárias para que elas cheguem a um consenso em tempo hábil, ou, após algumas iterações, a rainha pára a negociação e julga o resultado final da classificação para aquela entrada. As operárias são então desocupadas para atender a qualquer outra rainha e a rainha retorna o resultado final encontrado e se desincumbe daquela instância,;

6. Caso não haja rainhas que coordenam a extração livres, nascem novas rainhas coordenadoras.

7. A rainha responsável por coordenar a extração que estiver livre observa se o ambiente recebeu como resposta da classificação que a instância pertence à classe *esteganografada*, e, caso encontre uma instância desse tipo, incumbe-se de coordenar a extração da imagem representada pela instância. Caso não encontre nenhuma instância que precise passar pela extração, ela morre.

8. A rainha que se encarregar de extrair dados da imagens classificada como *esteganografada* (contendo conteúdo esteganografado), procura no ambiente por operárias especializadas em extrair dados para a técnica que se suspeita que foi empregada para esteganografá-los na imagem. Caso não consiga alocar uma quantidade de operárias extratoras satisfatória, ela procria novas operárias capazes de fazer tal tarefa e

repete este passo. Caso consiga alocar as operárias necessárias para fazer os ataques de extração em quantidade satisfatória, passa para o passo 9;

9. As operárias extratoras fazem diversos ataques (em paralelo) com o objetivo de extrair os dados que foram esteganografados na imagem descrita pela instância, e retornam o resultado de seu trabalho para a rainha.

10. A rainha verifica se uma das operárias extratoras conseguiu quebrar a esteganografia e, quando isto acontece, ela retorna os dados extraídos para o ambiente, desaloca as operárias do trabalho que estavam realizando e também se desincumbe da tarefa.

11. Caso todas as rainhas estejam mortas e não houver mais instâncias de teste que precisem de classificação nem instâncias classificadas que precisem de extração as operárias também morrem e o sistema termina. Caso contrário, o sistema volta para o passo 2.

De acordo com Booch (2000), a arquitetura de um sistema de software complexo pode ser descrita por cinco visões interligadas, são elas: visão de caso de uso, visão de projeto, visão de processo, visão da implementação, e visão da implantação. Portanto a nossa abordagem será apresentada a seguir através dessas cinco visões.

3.1 Visão de Caso de Uso

A Figura 3.1 mostra o diagrama de casos de uso do sistema. Os atores apresentados nesse diagrama são:

- *Monitor* – inicialmente refere-se aos usuários do sistema, que submetem um conjunto de arquivos monitorados para que os agentes a seguir trabalhem sobre tais arquivos; posteriormente a atividade de submeter buscas na internet para coletar dados para análise será feita por um agente chamado de Monitor ou Buscador;

- *Rainha (Agente Cliente)* – agente de software responsável por coordenar a atuação dos agentes classificadores; este agente atualmente tem o ciclo de vida delimitado somente enquanto a imagem que ele busca encontrar a classificação não é classificada; quando este agente não encontra uma quantidade suficiente

de agentes classificadores livres para trabalhar na sua imagem, ele então "procria" novos classificadores para a comunidade;

- *Operária (Agente Classificador)* – agentes de software que possuem classificadores variados; atualmente cada um é capaz de classificar se uma imagem é suspeita ou não de conter dados esteganografados; porém os agentes clientes não se satisfazem com a resposta de poucos classificadores, e estes negociam entre si para dar a resposta final;

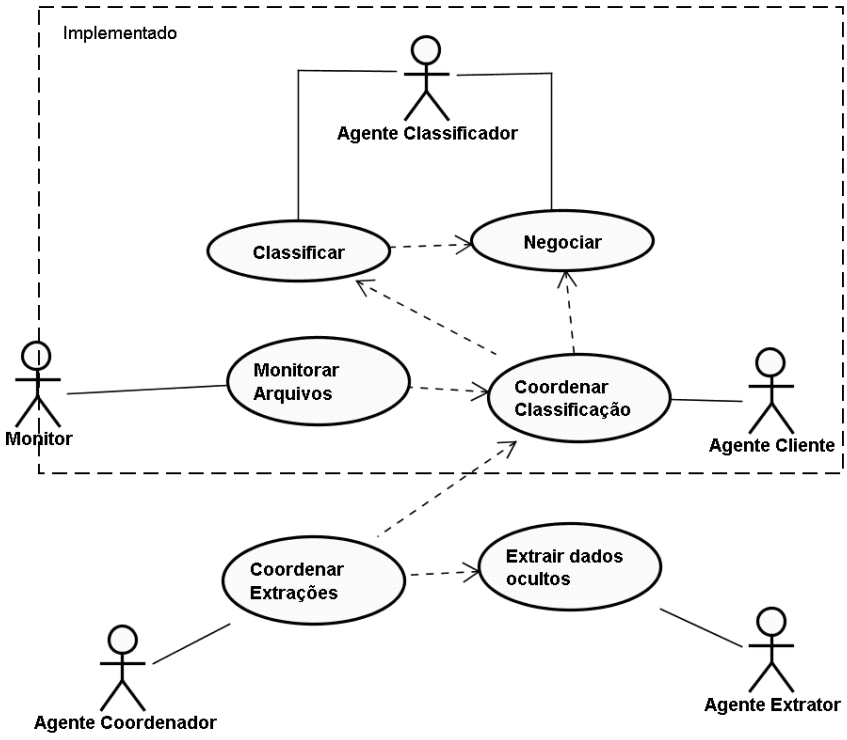

Figura 3.1 - Diagrama de Casos de Uso.

- *Rainha (Agente Coordenador)* – orquestram a atuação dos agentes extratores; cada agente coordenador é instanciado para coordenar a extração de dados de um arquivo em específico; após serem instanciados, estes agentes atribuem a tarefa de extração para os agentes indicados; caso não haja extratores livres estes agentes podem "procriar" novas instâncias de extratores;
- *Operária (Agente Extrator)* – agentes projetados para futuramente colher os dados suspeitos de servir como cobertura à esteganografias e atacá-los para extrair o conteúdo secreto.

Os casos de uso apresentados são:

- *Monitorar Arquivos* – tarefa atribuída aos monitores do sistema, consiste em selecionar um grupo de arquivos (atualmente apenas imagens) para que estes sejam atacados pelo sistema esteganalítico;
- *Coordenar Classificação* – tarefa realizada pelas rainhas (clientes); após realizar o pré-processamento dos arquivos colhidos na monitoração para obter as bases de treinamento, coordenam as atividades de classificação e negociação dos agentes classificadores;
- *Classificar* – atividade das operárias (classificadoras), consiste em analisar um arquivo e apontar quando este arquivo contém ou não dados ocultos; além disso deve indicar quais são os métodos de esteganografia que tem mais probabilidade de terem sido empregados;
- *Negociar* – após um grupo coordenado de operárias (classificadoras) darem suas opniões, elas comparam os seus resultados umas com as outras e emitem novas opniões – que são propostas de negociação de seu resultado;
- *Coordenar Extrações* – nesse caso de uso as rainhas (coordenadoras) do módulo extrator coordenam as atividades de extração, atribuindo a tarefa de extração para os agentes extratores mais indicados;
- *Extrair dados ocultos* – o próprio nome deste caso de uso já dá uma boa noção do que ele representa; consiste em atacar os arquivos classificados como esteganografados na tentativa de obter os dados ocultos dentro deles.

3.2 Visão de projeto

Dentro da visão de projeto, são tratados aspectos como o monitoramento e processamento, a extração especializada e a classificação negociada, descritos a seguir.

3.2.1 Monitoramento e Pré-Processamento

A atividade de monitoramento resulta em um conjunto de arquivos que então sofre o pré-processamento para ser atacado pelo sistema. Após os arquivos monitorados serem coletados, eles passam pela classe *estatísticas* que, utilizando-se das outras classes desse módulo, calcula os momentos estatísticos desses dados e gera uma instância para cada arquivo. Tais instâncias contêm os valores dos quatro momentos estatísticos (média, variância, assimetria e curtose), e quando se trata de dados de treinamento do sistema as instâncias também contêm os nomes das classes que estão sendo representadas.

O trecho de código do Quadro 3.1 mostra como estes momentos foram capturados em nossa abordagem. *Pixel* à p*ixel*, os dados são colhidos para os 6 canais de cores. Em seguida, os 4 momentos estatísticos são calculados para cada um dos canais. A forma como os momentos estatísticos é calculada pode ser vista no Quadro 3.2, os métodos *getMedia()*, *getVariancia()*, *getAssimetria()* e *getCurtose()* são métodos de encapsulamento para retornar à outra classe os valores dos momentos calculados.

```
/*
* Método para ler o conjunto de dados estatísticos de cada
imagem,
* retorna o conjunto de dados estatisticos...
*/
public static Dados lerEstatisticas(Image image, String
classe){
...
PixelGrabber p = new
PixelGrabber(image,0,0,largura,altura,pixels,0,largura);
try{
while(!p.grabPixels()){}//tenta pegar os pixels
} catch (InterruptedException ie ){}
for (int i =0; i< largura*altura; i++){
DadosPixel d = new DadosPixel(pixels[i]);
r[i] = d.getR();
g[i] = d.getG();
b[i] = d.getB();
h[i] = d.getH();
s[i] = d.getS();
v[i] = d.getV();
}
Estatistica e = new Estatistica(r);
dados.setMediaR(e.getMedia());
dados.setVarianciaR(e.getVariancia());
```

```
dados.setAssimetriaR(e.getAssimetria());
dados.setCurtoseR(e.getCurtose());
e = new Estatistica(g);
dados.setMediaG(e.getMedia());
dados.setVarianciaG(e.getVariancia());
dados.setAssimetriaG(e.getAssimetria());
dados.setCurtoseG(e.getCurtose());
...
e = new Estatistica(v);
dados.setMediaV(e.getMedia());
dados.setVarianciaV(e.getVariancia());
dados.setAssimetriaV(e.getAssimetria());
dados.setCurtoseV(e.getCurtose());
return dados;
}
```

Quadro 3.1. Trecho de código que ilustra como os dados são capturados no pré-processamento.

```
private void calcularMedia(float[] elementos){
   double soma = 0;
   for (int i=0; i < elementos.length; i++){
      soma = elementos[i];
   }
   media = soma/elementos.length;
}

private void calcularVariancia(float[] elementos){
   double soma = 0;
   calcularMedia(elementos);
   for (int i=0; i < elementos.length; i++){
      soma = elementos[i] - media;
   }
   variancia = Math.pow(soma,2)/(elementos.length-1);
}

private void calcularDesvioPadrao(float[] elementos){
   calcularVariancia(elementos);
   desvioPadrao = Math.sqrt(variancia);
}
```

```
private void calcularAssimetria(float[] elementos){
   double soma = 0;
   calcularDesvioPadrao(elementos);
   for (int i=0; i < elementos.length; i++){
      soma = elementos[i] - media;
   }
   assimetria = variancia*soma/Math.pow(desvioPadrao,3);
}

private void calcularCurtose(float[] elementos){
   double soma = 0;
   calcularAssimetria(elementos);
   for (int i=0; i < elementos.length; i++){
      soma = elementos[i] - media;
   }
   curtose = assimetria*soma/desvioPadrao -3;
}
```

Quadro 3.2. Trecho de código que ilustra como momentos estatísticos são calculados.

Quando um arquivo monitorado chega ao sistema, ele sofre o pré-processamento, transformando-se numa instância que passará então pela etapa de classificação. As principais classes utilizadas no pré-processamento para imagens digitais, ilustradas na Figura 3.2 são:

- *Instancia* – classe que guarda um vetor com os atributos dos dados estatísticos e o nome do arquivo;
- *DadosPixel* – pega pixel a pixel os valores das matrizes R, G e B, e a partir da transformação linear delas deduz as matrizes H, S e V;
- *Dados* – guarda os momentos estatísticos de média, variância, assimetria, e curtose das matrizes R, G, B, H, S e V, e, quando for o caso, a classe que será usada para treinamento dos métodos de aprendizado dos classificadores;
- *Estatística* – calcula a média, variância, desvio padrão, assimetria e curtose de um vetor de elementos;
- *Estatísticas* – a partir de uma lista de arquivos, gera uma base de dados com instâncias preparadas para treinamento e teste dos classificadores.

3.2.2 Classificação Negociada

Depois do pré-processamento, os dados monitorados passam pela classificação negociada. Nesta etapa cada instância é atribuída a um *Agente Cliente* que por sua vez coordena a classificação, que é efetuada por um conjunto de *Agentes Classificadores*. Os agentes comunicam-se entre si através de suas interfaces de conexão com a classe *Mensageiro*, que funciona como um *blackboard* para a troca de mensagens entre os agentes. O protocolo de troca de mensagens, representadas pela classe Mensagem, apesar de possuir sintaxe própria é inspirado em algumas das performativas básicas reservadas do KQML (Finn 97), como remetente, destinatário, assunto.

Atualmente os classificadores utilizados são diversas instanciações da classe *Arvore*, que implementa uma árvore de decisões. Porém, as instâncias de *Arvore* são diferenciadas uma das outras porque cada uma delas possui bases de treinamento e testes únicas. Essas bases são obtidas através de sorteios aleatórios dos dados inicialmente indicados para treinamento. As bases de treinamento e teste não possuem ocorrências da mesma instância.

3.2.3 Extração Especializada

Este módulo foi projetado para possuir recursos que o habilitem a separar os dados ocultos de suas coberturas. Sua principal classe é o *Agente Extrator*, cujas instâncias devem implementar diferentes utilitários especializados em atacar métodos específicos de esteganografia. Tais ataques podem ser, por exemplo, métodos de tentativa e erro exaustivos para encontrar as chaves que revelarão os dados. A coordenação destas agentes operárias também é efetuada por uma rainha, através do quadro de mensagens.

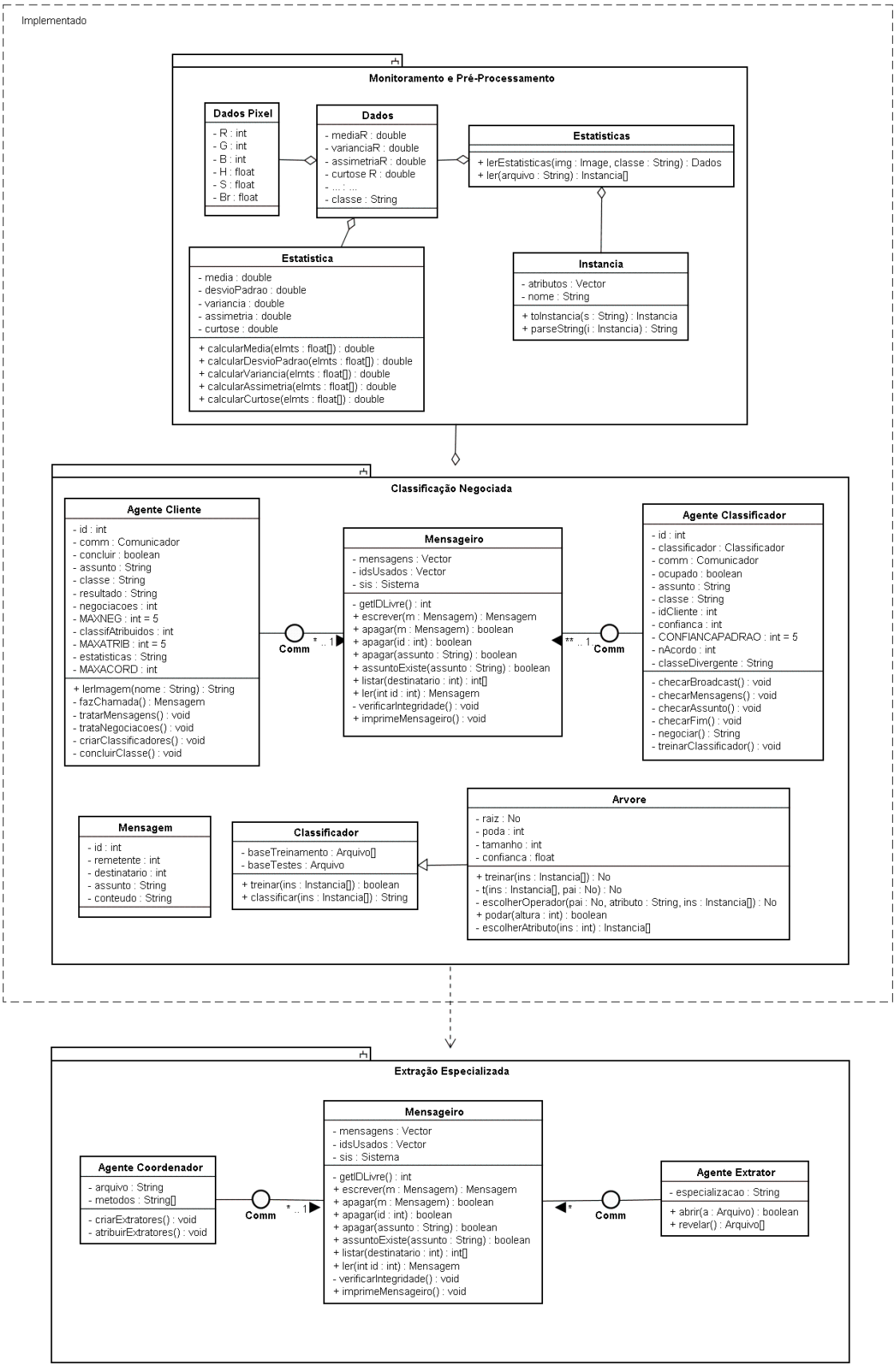

Figura 3.2 – Diagrama de Classes.

3.3 Visão de Processo

Todas as linhas de execução criadas são coordenadas em uma linha principal, que é a instância da classe Sistema. Cabe ao sistema iniciar os *Agentes Clientes* e os *Agentes* Coordenadores. Quando estes agentes requisitam novos *Agentes Classificadores* ou *Agentes Extratores*, a classe Sistema as instancia. Os mensageiros e os primeiros classificadores e extratores também são iniciados automaticamente pelo sistema. Dessa forma as tarefas são distribuídas através das *threads* exibidas na Figura 3.3.

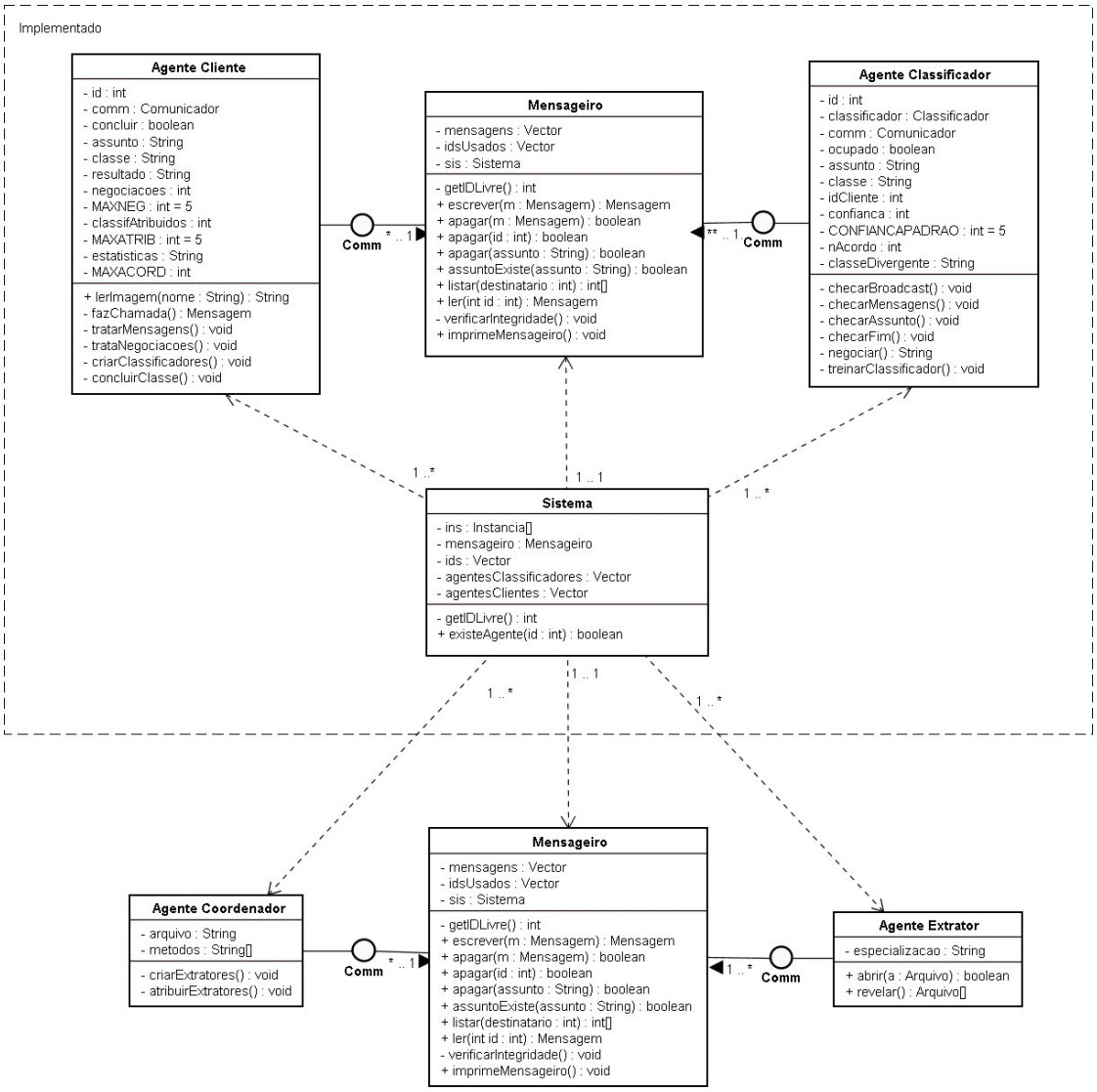

Figura 3.3 – Diagrama de Classes dos Processos.

3.4 Visão da Implementação

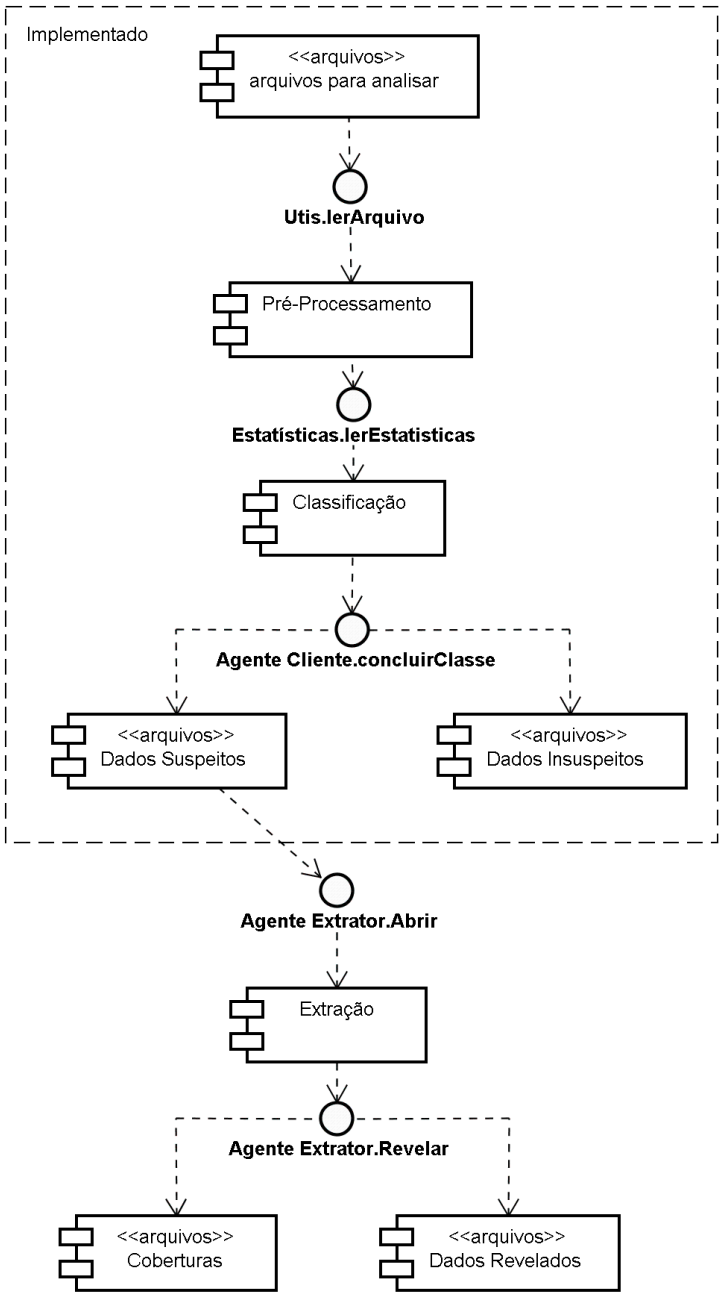

Figura 3.4 – Diagrama de Componentes.

A arquitetura do sistema proposto consiste de algumas etapas, que podem ser vistas na Figura 3.4:

1. arquivos monitorados são analisados no pré-processamento, que gera uma instância de dados estatísticos para cada arquivo;
2. cada dado que entra no módulo de classificação é tratado por uma rainha

(coordenadora), que faz a requisição de classificação para esse dado e instancia novas operárias (classificadoras) se for necessário; nesse módulo alguns agentes podem ser especializados para um dado tipo de mídia, ou para um dado tipo de técnica de esteganografia e outros agentes podem implementar métodos de esteganálise universal;

3. os dados, já classificados, são separados e os que apresentarem suspeita de possuir dados ocultos passam para o módulo de extração;
4. o módulo de extração consiste de um grupo de agentes especializados; cada agente desse módulo é composto de um componente especializado para extrair dados que foram ocultos por uma técnica de esteganografia específica;
5. por fim, os dados que conseguiram ser extraídos são revelados pelo sistema.

3.5 Visão da Implantação

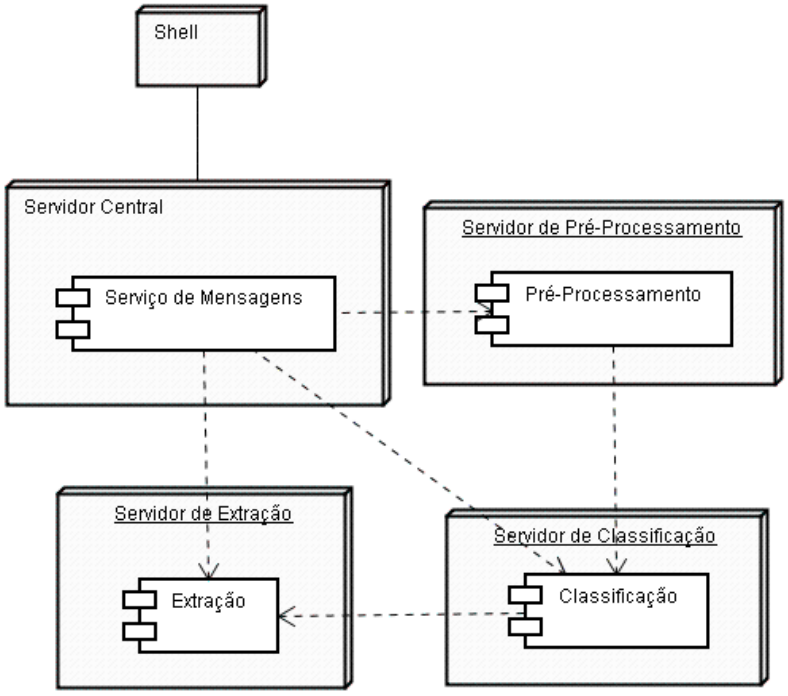

Figura 3.5 – Diagrama de Implantação.

Atualmente, a principal interface do usuário com o sistema é o *shell* ou linha de comando. Ele chama o sistema, conectando-se aos serviços centrais. O sistema pode ser totalmente distribuído em rede graças ao serviço centralizado de mensagem, e à implementação em *threads* deste e dos serviços de pré-processamento, classificação, e extração. O sistema pode ser implantado conforme o diagrama da Figura 3.5.

3.6 Modelo de Comunicação

O modelo de comunicação utilizado para coordenar os agentes baseia-se em um *blackbox*, um quadro de avisos onde os agentes anunciam o que estão tratando em relação a cada assunto, implementado na classe *Mensageiro* descrita anteriormente. Os agentes que exercem a função de *control shell* (ver seção 2.2.) são as rainhas. A Figura 3.6 mostra um exemplo de interações no quadro de comunicação do sistema. O primeiro campo do protocolo indica o número da mensagem; o segundo campo indica o agente que está mandando a mensagem; o terceiro é o destinatário; o quarto campo é o assunto (a imagem analisada); e o último campo é o corpo da mensagem.

```
============================================================
======================== MENSAGEIRO ========================
============================================================
1 1 -1 esteganografada/s136.jpg CRIAR=10
2 1 0 esteganografada/s136.jpg CHAMADA
3 1 -1 esteganografada/s136.jpg CRIAR=1
============================================================
============================================================
```

Figura 3.6 – Exemplo de interações no quadro de comunicação.

Destinatário 0 significa que a mensagem é em broadcast, -1 significa que a mensagem é uma interação com o ambiente, e demais números são os números dos agentes alvo. Note que no campo assunto desses exemplos, em alguns momentos aparece a palavra *normal* e em outros *esteganografada*, ambas seguidas de "/" e do nome do arquivo. Nesse caso *normal* e *esteganografada* são os diretórios a partir do sistema onde as imagens encontram-se, mas estas imagens estão já separadas nessas pastas apenas para atender a questões dos experimentos como o cálculo posterior do percentual de acerto. O corpo pode conter as mensagens:

- CHAMADA – mensagens enviadas em *broadcast* por um Agente Cliente, requisitando que os Agentes Classificadores respondam se estão presentes no ambiente; esta mensagem serve para o Agente Cliente saber se existem ou não Agentes Classificadores o suficiente para cooperar e negociar no processo de classificação;
- CRIAR=*N* – quando, através da chamada, o Agente Cliente percebe que precisa de mais Agentes Classificadores no ambiente para ajudá-lo, ele envia esta mensagem ao ambiente indicando que deseja que *N* Agentes Classificadores sejam criados;
- PRESENTE – ao ouvir uma CHAMADA de um Agente Cliente, todo Agente Classificador responde a este Agente Cliente que está presente no ambiente;
- CLASSIFICAR=*entrada* – ao tomar ciência de que um Agente Classificador está presente, o Agente Cliente envia esta mensagem para ele requisitando que o mesmo classifique a entrada;
- ACEITO – caso o Agente Classificador esteja desocupado ele envia esta mensagem ao ser requisitado para realizar uma classificação, e efetua este processo;
- OCUPADO – esta mensagem é enviada de um Agente Classificador para um Agente Cliente, quando este Agente Classificador já está trabalhando em outra entrada diferente (negando a requisição de uma outra classificação);
- CLASSE=*classe* – ao realizar a classificação, o Agente Classificador envia esta mensagem para o Agente Cliente, indicando qual foi a classe encontrada;
- ACORDO=*classe nIteração* – quando dois ou mais Agentes Classificadores, que estão trabalhando em uma entrada, encontram classes diferentes, eles entram em um processo de negociação que resulta em novas classes que podem ser encontradas por *nIteração* vezes; ao entrarem em acordo o Agente Cliente aceita a classe acordada e finaliza a tarefa, liberando os classificadores para outros clientes.

3.7 Negociação Coordenada

Os agentes são coordenados conforme a metáfora das abelhas M. Bicolor citada no Capítulo 2. Nessa heurística, podemos comparar os módulos de classificação e de extração com duas colméias. No módulo classificador, cada instância da rainha (cliente) funciona como uma abelha rainha que além de criar, coordena qualquer operária (classificadora) sem fazer distinção se os classificadores são de sua criação ou não. Da mesma forma as rainhas (coordenadoras) agem com relação às operárias extratoras.

Quando uma rainha (agente cliente) precisa classificar um arquivo, ela faz uma chamada no sistema para convocar as operárias (*agentes classificadores*) que puder. Então coordena as atividades de classificação desse grupo de operárias classificadoras e quando há discordância no resultado desses agentes, o cliente também coordena sua negociação.

Os agentes classificadores respondem a chamada, em seguida recebem um pedido de classificação e respondem se aceitam o pedido ou se estão ocupados. Ao aceitar o pedido de classificação de um problema, todos os agentes que estiverem trabalhando nesse pedido farão suas classificações individuais e retornarão um resultado. Como estes resultados podem diferir entre os agentes, e como estes agentes podem negociar com seus colegas para que um deles mude de resultado, chamamos estes resultados de ***opiniões***.

A negociação baseia-se no protocolo de concessão monotônica. A cada iteração da negociação as operárias (*classificadoras*) avaliam suas respostas em comparação com as respostas do grupo que está trabalhando no mesmo problema; e hora concedem mudando sua opinião de classificação, hora insistem em se reafirmar perante opiniões divergentes. A equação 3.1 descreve como essas concessões são feitas.

Equação 3.1

$$confiança(x) = \begin{cases} confiança(x) + ni * confiança(x) / q\_interações \\ confiança(x) - ni * confiança(x) / q\_interações \end{cases}$$

Onde: q_interações é a quantidade total de interações, *ni* é o número de cada iteração, que varia de 1 até q_interações; *confiança* tem um valor inicial entre 0 e 5, variando conforme o grau de confiança de cada classificador.

Enquanto a confiança for positiva o agente não muda sua resposta; quando a confiança é negativa, o agente cede sua resposta para a resposta do outro.

Após algumas iterações, empiricamente escolhemos 5, se as operárias não chegarem a um consenso, a rainha (cliente) encerra a negociação e escolhe uma das classes mais indicadas.

3.8 Detalhes Internos dos Agentes

Nesse sistema, cada um dos agentes é um processo em execução, e possui um módulo interno chamado de *comumicador*, que é sua interface com o quadro-negro que representa o ambiente. A descrição das características individuais de cada agente pode ser vista a seguir. As rainhas dividem-se entre as tarefas de coordenar a classificação ou coordenar a extração dos dados esteganografados, efetuadas pelas operárias. As operárias são especializadas entre as tarefas de classificação e extração.

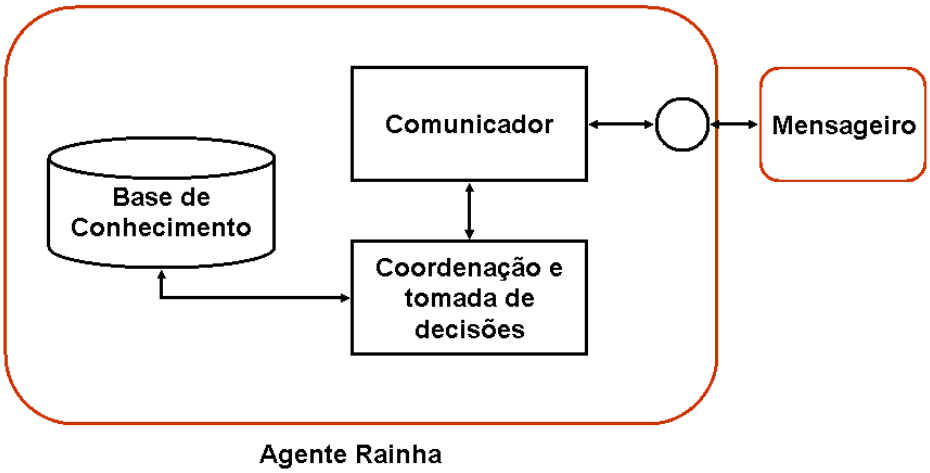

Figura 3.7 – Arquitetura interna dos agentes do tipo Rainha.

Rainha (Agente Cliente e Agente Coordenador) – neste texto chamamos dois tipos diferentes de agentes por *rainha*. Um deles tem o codinome *Agente Cliente* e o

outro *Agente Coordenador*, porém ambos coordenam o trabalho das operárias. A diferença entre esses agentes é que a rainha "cliente" coordena as atividades das operárias classificadoras, e a rainha "coordenadora" coordena as atividades das operárias extratoras, porém sua arquitetura interna é a mesma, (Figura 3.7). Os componentes internos de ambos os tipos de agentes rainha são:

- *Base de conhecimento* – guarda informações sobre as operárias que a rainha está coordenando, o problema (exemplo) que está sendo trabalhado por este grupo, e os resultados das interações passadas desse agente com os demais;
- *Coordenação e tomada de decisões* – módulo desse agente aonde as mensagens que chegam para ele são analisadas e ações (mensagens para as operárias ou procriação) são tomadas de acordo com os objetivos do mesmo;
- *Comunicador* – como descrito no início dessa seção, exerce o papel de interface entre este agente e o quadro-negro (mensageiro) que simboliza o ambiente; a cada instante o comunicador analisa quando novas mensagens são direcionadas para este agente, e passa essas mensagens para o módulo de coordenação e tomada de decisões; também sendo responsável por entregar ao mensageiro as mensagens que este outro módulo gera.

Operária Classificadora – as operárias que assumem o papel de classificação são tem como objetivo atender às requisições das rainhas para classificação de imagens entre as classes *normal* ou *esteganografada*, que indica para as rainhas quando uma imagem é ou não suspeita de conter dados esteganografados. Para que uma imagem seja classificada, o grupo de operárias classificadoras que estiver trabalhando sobre esta imagem deve, quando necessário, negociar seus resultados para chegar a um consenso. Os componentes internos de uma operária classificadora são ilustrados na Figura 3.8, e são:

- *Base de conhecimento* – guarda informações sobre a rainha a qual a operária está temporariamente subordinada, o exemplo que está sendo classificado, o desempenho do algoritmo de aprendizado de máquina

utilizado, e os resultados das interações passadas desse agente com os demais para a realização da tarefa de classificação atual;

- *Negociação e tomada de decisões* – módulo desse agente aonde as mensagens que chegam para ele são analisadas e ações são tomadas de acordo com os objetivos do mesmo; tais ações pode ser o início da classificação, troca de mensagens com a rainha ou proposta de negociação para as demais operárias do grupo;
- *Algoritmo de Classificação* – neste módulo é implementado um dos algoritmos de aprendizado de máquina citados no capítulo 2, quando a classe de um desses agentes é instanciada pelo sistema este algoritmo é treinado de acordo com um conjunto de dados para treinamento e testes que o sistema possui; o resultado do desempenho deste algoritmo é guardado na base de conhecimento e influencia o quanto este agente é seguro sobre o resultado de sua classificação no momento da negociação com as demais operárias;
- *Comunicador* – idem descrição do comunicador para as rainhas.

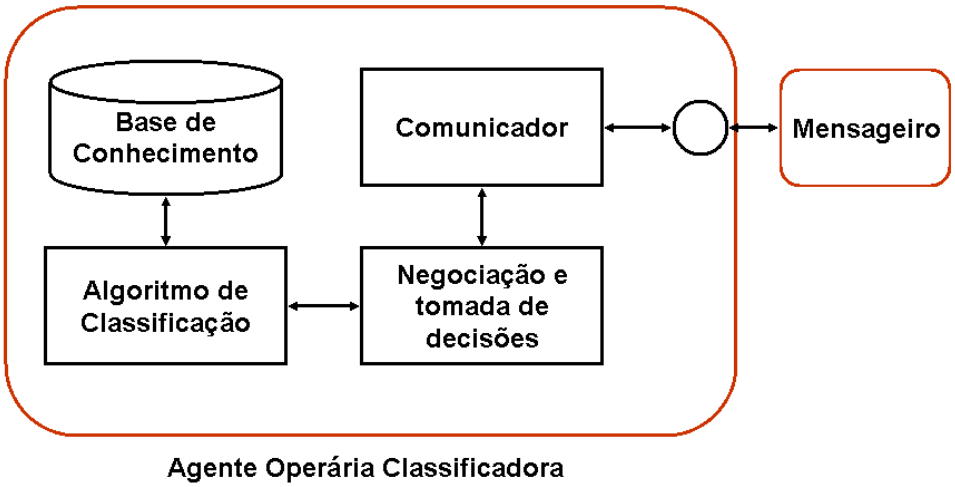

Figura 3.8 – Arquitetura interna dos agentes do tipo Operária Classificadora.

Operária Extratora – são os agentes que têm como objetivo extrair os dados das imagens que foram classificados com a classe *esteganografada*. Estes agentes implementam algoritmos de ataque à imagens contendo dados esteganografados. Os componentes internos (Figura 3.9) da Operária Extratora são:

- *Base de conhecimento* – guarda informações sobre a rainha a qual a operária está temporariamente subordinada, a imagem que está sendo atacada, o tipo de algoritmo de esteganografia que este agente ataca, e os resultados das interações passadas desse agente sua rainha para a realização da tarefa de classificação atual;
- *Tomada de decisões* – neste módulo as mensagens que chegam para ele são analisadas e ações são tomadas de acordo com os objetivos do mesmo; tais ações basicamente são o ataque de extração, a troca de mensagens com a rainha, e a análise dos dados obtidos a cada tentativa de extração para verificar se tais dados têm alguma consistência ou se são dados aleatórios;
- *Algoritmo de Extração* – neste módulo pode ser implementado um dos algoritmos de extração de informações esteganografadas existentes, ou pode ser implementado um algoritmo que tente quebrar a chave de uma técnica de esteganografia através da tentativa e erro; este algoritmo é executado até que o módulo de tomada de decisões interprete o resultado dos dados retornados por ele de forma satisfatória;
- *Comunicador* – idem descrição do comunicador para as rainhas.

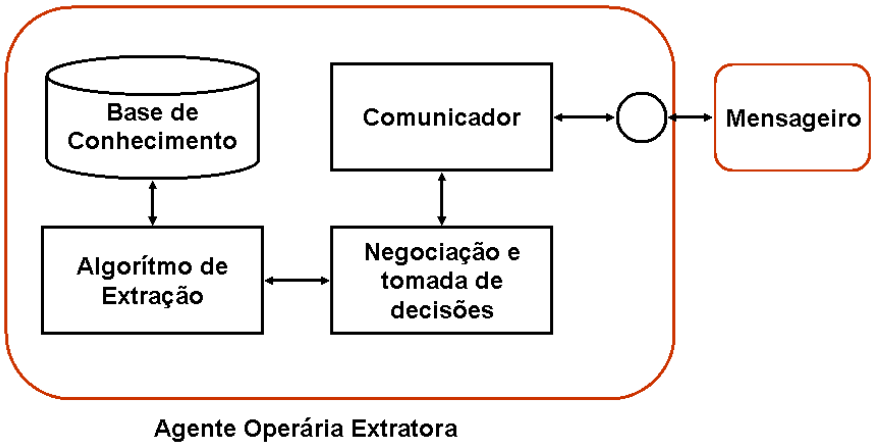

Figura 3.9 – Arquitetura interna dos agentes do tipo Operária Extratora.

Capítulo 4.

Experimentos e Resultados

"Todo o conhecimento genuíno tem origem na experiência direta." Mao Tse-Tung

Realizamos dois tipos de experimentos, no tocante ao seu objetivo. O primeiro conjunto de experimentos foi realizado para otimizar a quantidade de atributos usados para treinar os algoritmos de aprendizado de máquina. O segundo conjunto foi realizado para avaliar o desempenho do sistema. Nas seções a seguir, discutiremos como foi realizado cada tipo de experimento e quais foram os resultados obtidos.

4.1 Descrição

As duas etapas de experimentação foram realizadas seguindo a metodologia de planejamento de experimentos apresentada por Cobb (1997). Tal metodologia consiste em tomar três decisões:

1. Que medida tomar? (Que métricas serão usadas para avaliar a resposta do sistema?)
2. Que condições deve-se estudar? (Qual tratamento será dado ao problema, que hipóteses serão testadas com o experimento?)
3. Que material experimental usar? (Que unidades serão estudadas?)

Descrevemos a aplicação de tal metodologia a seguir.

4.1.1 Métricas

As respostas esperadas dos experimentos são certas medidas acerca da classificação realizada pelo SMA. Tais medidas são:

- Taxa de acertos (%) – o percentual de acertos que o sistema apresenta, ao ser submetido à base de dados de testes;
- Falsos positivos (%) – indica os alarmes falsos, ou o percentual de vezes em que o sistema classificou erroneamente arquivos sem mensagens esteganografadas

como se possuíssem tais mensagens;

- Falsos negativos (%) – indica o percentual de vezes em que o sistema não detectou a presença de dados esteganografados, quando estes existiam, ao ser submetido aos teste.

4.1.2 Condições de Estudo

O sistema é apresentado à um conjunto de dados de treinamento e à um conjunto de dados de teste. Os dados de treinamento são distribuídos aleatoriamente em novos conjuntos de treinamento não totalmente exclusivos, destinados ao treinamento dos agentes classificadores. Os dados de teste são distribuídos entre os agentes clientes para que estes coordenem as atividades de classificação dos agentes classificadores em torno destes dados. O objetivo deste estudo é obter as taxas citadas anteriormente e compará-las com as taxas encontradas em experimentos semelhantes apresentados na literatura para verificar se a nossa abordagem apresenta ou não ganhos em relação ao estado da arte.

4.1.3 Material Estudado

Devido à variedade de dados que o sistema proposto pode abranger; restringimos nossos experimentos atuais apenas às imagens digitais. Os experimentos são realizados com 300 imagens de paisagens, interiores, animais, construções, pessoas, plantas e alimentos. As imagens possuem dimensão média de 800x600 *pixels*. Uma metade aleatoriamente selecionada das imagens permaneceu inalterada, enquanto a outra ganhou dados ocultos em seu conteúdo através da técnica de esteganografia **JPHide/JPSeek** (Lathan 2006). Os dados ocultos ocupam cerca de 10% do tamanho dos dados de cobertura. A partir desse novo conjunto de imagens, foi realizado o pré-processamento para extrair os dados que formam as bases de treinamento e testes.

4.2. Experimentos Preliminares

No primeiro grupo de experimentos utilizamos o pacote de algoritmos de aprendizado de máquina *Weka (2007)*, para submeter o conjunto completo de dados que extraímos das imagens a alguns desses algoritmos. Em seguida pré-selecionamos, por apresentarem menor complexidade para extração, alguns dos atributos utilizados nos trabalhos citados no capítulo 2 e em outros trabalhos de segmentação de imagem. Algumas permutações desses atributos foram submetidas aos algoritmos de aprendizado de máquina: Árvores de Decisão, K-nn, e MLP. A permutação que se saiu melhor utilizava os atributos dos quatro primeiros momentos estatísticos (média, variância, assimetria e curtose) de alguns canais de cores, tanto da matriz RGB quanto da HSV, portanto entendemos que é melhor utilizar como atributos da nossa base os momentos estatísticos dos seis canais de cores das duas matrizes citadas.

Pelo que constatamos nos trabalhos referenciados, e nos experimentos, esses dados estatísticos apresentam uma sensibilidade importante para classificação de diversos tipos de problemas envolvendo análise e segmentação de imagens – inclusive a esteganálise.

4.3. Experimentos Sobre o Sistema

Após o desenvolvimento do sistema como um todo, foram realizados outros experimentos, seguindo o passo a passo apresentado anteriormente, visando sua validação. Nesses experimentos realizados, o sistema apresentou uma taxa de acertos de 82,37%, com falsos positivos de 9,62% e falsos negativos de 8,01%.

A seguir ilustramos, passo a passo, como o sistema evolui até chegar a uma primeira classificação. Tomamos como exemplo a Figura 4.1, que ilustra um arquivo contendo um texto (4.1.a) sendo oculto em uma imagem de cobertura (4.1.b) e gerando o objeto esteganográfico (4.1.c). Aparentemente, a imagem (objeto esteganográfico) gerada não difere da imagem original utilizada como cobertura, porém, através da detecção de pequenas distorções em seus dados estatísticos (sem se conhecer a imagem de cobertura), é possível fazer a detecção da presença dos dados ocultos, como será visto.

Figura 4.1 – Exemplo de esteganografia: a) texto a ser oculto, b) imagem de cobertura, c) objeto esteganográfico.

As mensagens ilustradas na seqüência retratam algumas iterações do sistema para realizar a esteganálise da imagem da Figura 4.1. O formato dessas mensagens é explicado na seção 3.6. E o significado desses resultados será discutido em seguida.

```
 ============================================================
======================== MENSAGEIRO ========================
============================================================
2 1 0 esteganografada/s136.jpg CHAMADA
3 1 -1 esteganografada/s136.jpg CRIAR=1
============================================================
============================================================
```

Figura 4.2 – Apresentação de um problema para as operárias.

A Figura 4.2 mostra o momento em que o Agente1 Cliente (ou Rainha) faz uma chamada para que os Agentes Classificadores (ou Operárias) realizem uma tarefa. A Figura 4.3 mostra um momento posterior onde o Agente2 Classificador responde que está presente, e o agente1, ainda insatisfeito, pede ao ambiente que crie mais um classificador.

Posteriormente, a Figura 4.4 mostra o momento onde o agente1 pede ao agente2, que anteriormente havia recebido a chamada, que faça a tarefa de classificação de uma entrada. Esta entrada aqui será chamada de *Entrada A*, e corresponde ao arquivo `esteganografada/s136.jpg` e aos dados estatísticos listados na Tabela 4.1. A Figura 4.5 mostra o momento em que o Agente2 termina sua classificação individual dessa entrada.

```
============================================================
======================== MENSAGEIRO ========================
============================================================
1 2 1 esteganografada/s136.jpg PRESENTE
2 1 0 esteganografada/s136.jpg CHAMADA
3 1 -1 esteganografada/s136.jpg CRIAR=1
============================================================
============================================================
```

Figura 4.3 – Operária responde chamada.

```
============================================================
======================== MENSAGEIRO ========================
============================================================
3 1 -1 esteganografada/s136.jpg CRIAR=1
4 1 2 esteganografada/s136.jpg CLASSIFICAR=1.25E-5,
7.499984374999998E-5, 692.8196013393389, 479996.00000000035,
1.25E-5, 7.499984374999998E-5, 692.8196013393389,
479996.00000000035, 1.25E-5, 7.499984374999998E-5,
692.8196013393389, 479996.00000000035, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 4.901960880185167E-8,
1.1534001796795082E-9, 692.8196013393383,
479995.99999999977,
============================================================
============================================================
```

Figura 4.4 – Rainha solicita classificação.

Matriz	Média	Variância	Assimetria	Curtose
R	1.25E-5	7.499984374999998E-5	692.8196013393389	479996.00000000035
G	1.25E-5	7.499984374999998E-5	692.8196013393389	479996.00000000035
B	1.25E-5	7.499984374999998E-5	692.8196013393389	479996.00000000035
H	0.0	0.0	0.0	0.0
S	0.0	0.0	0.0	0.0
V	4.901960880185167E-8	1.1534001796795082E-9	692.8196013393383	479995.99999999977

Tabela 4.1 – Dados estatísticos da entrada esteganografada/s136.jpg.

```
============================================================
======================== MENSAGEIRO ========================
============================================================
4 1 -1 esteganografada/s136.jpg CRIAR=1
1 2 1 esteganografada/s136.jpg PRESENTE
2 2 1 esteganografada/s136.jpg ACEITO
3 2 1 esteganografada/s136.jpg CLASSE=normal
============================================================
============================================================
```

Figura 4.5 – Operária classifica entrada.

Da mesma forma que o agente2 interagiu com o agente1, o agente3 interage com o agente1. Do resultado dessa iteração, a classe encontrada pelo agente2 foi a classe *normal*, enquanto que o agente3 detectou a classe *esteganografada.* O que desencadeou todo o processo de negociação que pode ser visto na Figura 4.6.

```
============================================================
======================== MENSAGEIRO ========================
============================================================
4 3 1 esteganografada/s136.jpg ACORDO=esteganografada 2
1 2 1 esteganografada/s136.jpg ACORDO=normal 2
2 3 1 esteganografada/s136.jpg ACORDO=esteganografada 3
3 2 1 esteganografada/s136.jpg ACORDO=normal 3
6 3 1 esteganografada/s136.jpg ACORDO=esteganografada 4
5 2 1 esteganografada/s136.jpg ACORDO=esteganografada 4
============================================================
```

Figura 4.6 – Acordo entre duas operárias para classificação final de uma entrada.

Nesta figura podemos acompanhar todo o processo de negociação entre o agente2 e o agente3 até o momento onde ambos concordaram que a *Entrada A* (imagem s136.jpg) deve ser classificada como *esteganografada*.

4.4. Análise dos Resultados

Em comparação com as taxas de acerto encontradas na literatura (entre 20% a 80% para mensagens ocultas com 10% do tamanho das coberturas), as taxas de acerto (em torno de 82% para a mesma situação) encontradas em nossos experimentos indicam que o método de aprendizado utilizado (as árvores de decisão), combinado com SMA, produziu resultados importantes para o problema em questão.

Melhorias na taxa de acerto de um algoritmo de aprendizado de máquina são comuns quando utilizados métodos de agrupamento desses algoritmos, porém, nem todos os algoritmos de AM podem ser melhorados com essas técnicas. O SMA abordado funcionou como um agrupamento de vários classificadores, melhorando assim o percentual de acertos do sistema. Alguns dos sistemas de esteganálise estudados utilizaram outras formas de agrupamento, mas não apresentaram resultados melhores que os nossos.

Isto é, apesar do desempenho de uma árvore única (em torno de 67% de acertos) ter melhorado com o uso do sistema multi-agentes apresentado (para 82%), o desempenho do sistema poderá ser consideravelmente melhor quando outros métodos usando diferentes de classificadores forem acrescentados aos diversos agentes classificadores. Assim,, já podemos ver tal ganho como algo positivo com o uso da arquitetura do SMA. Com o aumento da base de dados de treinamento e testes e com a adição de algoritmos diferentes de classificação, este percentual pode melhorar ainda mais, diminuindo os percentuais de falsos positivos e falsos negativos.

Capítulo 5.

Considerações Finais

"Vox audita perit littera scripta manet."

(A voz ouvida se perde, a letra escrita permanece.)

Neste trabalho, propomos uma abordagem para esteganálise de dados digitais. O método proposto detecta se um arquivo digital está ou não com alguma mensagem oculta em seu conteúdo. Como visto, a estrutura do sistema foi projetada para futuramente realizar por completo o processo de esteganálise, extraindo os dados protegidos. Desenvolvemos uma arquitetura flexível e autônoma o suficiente para atender aos requisitos de manipulação de grandes volumes de dados e adaptação às evoluções constantes das áreas envolvidas. Foi usada a abordagem multiagentes para a implementação do sistema. A estratégia de coordenação entre os agentes do sistema de esteganálise foi inspirada pelo modelo biológico das abelhas poligínicas. Portanto, em suma, no presente trabalho, desenvolvemos um SMA cuja estrutura é adequada para a resolução do problema de esteganálise em dados digitais.

O sistema proposto foi desenvolvido com agentes treinados para trabalhar com imagens digitais, usando as técnicas de esteganografia JPHide/JPSeek citada anteriormente. As tecnologias empregadas no projeto e implementação do sistema foram Jude Community (Change Vision Inc, 2006), para elaboração dos diagramas UML, e a programação feita em Java (Sun Microsystems Inc, 2006). As imagens foram coletadas no site de buscas Google (2006) e também foram tiradas por uma webcam Creative(R).

5.1 Contribuições

Embora o sistema tenha sido testado apenas com imagens digitais (cerca de 300 imagens foram usadas nos experimentos), o mesmo pode ser usado em dados digitais diversos. Como resultado, desenvolvemos uma arquitetura de sistema esteganalítico mais geral, que abranja todos os tipos de dados e técnicas esteganográficas. Pela

natureza dos dados estatísticos analisados, pudemos concluir que o sistema é universalmente sensível à detecção de presença de dados esteganografados (devido a sensibilidade às medidas estatística que tais dados apresentam).

Assim, as duas contribuições principais do presente trabalho são a arquitetura autônoma e adaptável do sistema esteganalítico universal, capaz de concorrer com ou superar os métodos de esteganálise e esteganografia atuais, e uma nova heurística de coordenação de sistemas multiagentes, inspirada pelas comunidades de abelhas poligínicas. Várias aplicações podem se beneficiar do presente sistema, incluindo áreas como economia, defesa e soberania nacional, bem como outras áreas que envolvam a necessidade de saber se há dados ocultos ou não em dados digitais, como perícia criminal e aplicações militares. Atualmente, estamos utilizando o sistema desenvolvido para identificar a presença de anomalias em dados digitais captados por sensores implantados em regiões da cabeça de cobaias (ratos) próximas do cérebro. Esta pesquisa, desenvolvida em parceria com o Instituto Internacional de Neurociência de Natal tem o intuito de confirmar a completude, pelas áreas do córtex cerebral, das imagens formadas durante o sono (sonho).

Ainda, como as áreas de aplicação da esteganografia coincidem com as áreas de aplicação da esteganálise, se a esteganografia é disponível para uso pessoal, admite-se que a esteganálise também possa ser utilizada por motivos pessoais (tais como a curiosidade).

5.2 Conclusões

Com a realização do presente o trabalho, pudemos concluir que a arquitetura autônoma e adaptável de sistema esteganalítico universal é capaz de concorrer ou superar os métodos de esteganálise e esteganografia atuais, pelos indicadores dos experimentos realizados. A arquitetura demonstra ser promissora e competitiva com as outras encontradas atualmente, e tem o potencial de superar ainda mais seu resultado a partir de otimizações a serem implementadas em trabalhos futuros (vistas a seguir).

A nova heurística de coordenação de sistemas multi-agentes apresentada aqui é uma heurística que pode ser utilizada não somente para a arquitetura desses sistemas, mas também para a resolução de problemas computacionais que demandem a coordenação da divisão de tarefas entre elementos.

Além disso, é possível concluir que adicionando agentes especializados em outras técnicas (específicas ou universais) e em outras mídias, a arquitetura pode atingir o objetivo de realizar a esteganálise universal de (quaisquer) dados digitais.

5.3 Planos Futuros

Os planos futuros são de ampliar a gama de classificadores, as técnicas de esteganografia e mídias alcançadas pelo sistema; e a construção do módulo de extração dos dados. Para evitar o trabalho de monitoramento, designado para pessoas; o ator monitor, introduzido na Figura 4.1.1, deverá ser implementado como uma entidade de software autônoma em futuras versões. O próximo passo é desenvolver o módulo de extração, que não foi desenvolvido no presente trabalho devido à complexidade envolvida para realizar tal tarefa.

Novos experimentos serão realizados com uma quantidade maior de arquivos. Atualmente, além dois trabalhos com dados cerebrais, já estamos preparando 1000 imagens para as próximas atividades. Em experimentos futuros, pretendemos utilizar 10.000 imagens, entre fotos coloridas, em escalas de cinza e ilustrações (que não foram usadas neste trabalho). Heurísticas como a “leave-one-out” serão utilizadas nos testes, visando determinar meios de melhorar a eficiência do método.

Ainda, apesar de modelar o problema de forma distribuída, usando diversas *threads*, o sistema ainda precisa de adaptações para executar as diversas instâncias dessas linhas de forma efetivamente distribuída em rede.

Glossário

AD – Árvore de Decisão, algoritmo de AM que expressa um processo de tomada de decisão através de uma estrutura hierárquica de decisões intermediárias, com o objetivo de classificar problemas.

AM – Aprendizado de Máquina, parte da IA que estuda algoritmos capazes de construir conhecimento de forma automática.

DCT – *Discrete Cosine Transform* (transformada discreta dos cossenos), transformada matemática que converte uma função numa série de cossenos de amplitudes e fases diferentes, que somados resultam num sinal equivalente ou aproximado ao da função.

DFT – *Discrete Fourrier Transform* ou Transformada Discreta de Fourrier, que transforma uma função em uma série de senos e cossenos de amplitudes e fases diferentes, que somadas resultam num sinal equivalente ou aproximado ao da função.

DWT – *Discrete Wavelet Transform* (transformada discreta de *wavelets* ou ondaletas), que aproxima uma função em várias resoluções diferentes, de acordo com a quantidade de elementos de uma série de funções Wavelet base (existem diversas variações dessas funções, como por exemplo a função "sombrero", cujo sinal tem o formato do chapéu típico mexicano).

IA – Inteligência Artificial, campo da computação que estuda métodos de se construir sistemas computacionais que apresentem comportamento inteligente.

JPG – método comum de compressão de imagens.

LDA – *Linear Discriminant Analysis*, também conhecido como classificante linear de Fisher, dentre outros nomes, é uma técnica de classificação que consiste em encontrar uma direção num subspaço do espaço original onde as classes estejam separadas o melhor possível.

LSB – *Less Significative Bit* ou *bit* menos significativo; os algoritmos de esteganografia por substituição LSB usam os bits menos significativos para colocar os dados ocultos.

PCM – Protocolo de Concessão Monotônica, protocolo de negociação onde os participantes negociam interativamente até encontrarem a solução de menor custo para ambos.

SMA – Sistema Multi-Agentes, sistema onde diversos *agentes* de *software* interagem de forma autônoma entre si e com um ambiente.

Sniffing – (farejamento) é uma técnica de espionagem que consiste em interceptar o tráfego de dados em num segmento de rede de computadores.

SVM – Support Vector Machines, que separam classes através de hiperplanos construídos na fase de treinamento.

Wavelet – função básica da transformada de ondaleta (ver DWT), definida *ad hoc* para aproximar uma determinada função através de uma série composta por elementos dessa função *wavelet*.

Referências

(Alves 2002) ALVES, Rex Nazaré. Inteligência e Sustentação dos Interesses Nacionais Economia, Ciência e Tecnologia e o Estado in *Seminário Atividades de Inteligência no Brasil: Contribuições para a Soberania e a Democracia.* Agência Brasileira de Inteligência, Brasília, nov. 2002.

(Aponte 2003) APONTE, Olga Inés Cepeda: *Poliginia e monoginia em Melipona bicolor (Apidae,Meliponini): do coletivo para o individual*. Tese de Doutorado, Instituto de Biociências, Universidade de São Paulo, 2003.

(Artz 2001) ARTZ, Donovan. *Digital Steganography: Hiding Data within Data*. IEEE Internet Computings, [s.l.], v.5, n.3, mai.-jun. 2001.

(Barrios-Aranibar 2005) BARRIOS-ARANIBAR, Dennis; ALSINA, Pablo J.. Recognizing Behaviors Patterns in a Micro Robot Soccer Game in *proceedings of the fifth International Conference on Hybrid Intelligent Systems* - HIS05, Rio de Janeiro, Brasil, 11/2005.

(Boch 2000) BOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar; UML – Guia do usuário. Ed. Campus. Rio de Janeiro:2000.

(Bradshaw 1997) BRADSHAW, Jeffrey M. (ed). *Software Agents*. The MIT Press. 1997. ISBN 0-262-52234-9.

(Burges 1998) BURGES, Christopher J. C. : A Tutorial on Support Vector Machines for Pattern Recognition in *Data Mining and Knowledge Discovery*, vol 2, 1998. Kluwer Academic Publishers, Boston.

(Chandramouli 2004) CHANDRAMOULI, R.; SUBBALAKSHMI, K.P.: Current Trends in Steganalysis: A Critical Survey, *Invited session on Multimedia Security, The Eighth International Conference on Control, Automation, Robotics and Vision, ICARCV 2004*, December 2004.(invited paper)

(Change 2006) CHANGE Vision Inc. *System Design Tool – JUDE: UML, ER, CRUD, Flowchart and Mind Map.* Disponível na Internet: http://jude.change-vision.com/jude-web/product/community.html . 31 out 2007.

(Cobb 1997) COBB, George W. *Introduction to design and analysis of experiments.* Ed. Springer. 1997. ISBN 0-387-94607-1.

(Dietterich 2000) DIETTERICH, T.G.: Ensemble methods in machine learning. In *Lecture Notes in Computer Science: Multiple Classier Systems*, Cagliari, Italy, 2000.

(Finin 1997) FININ, Tim; LABROU, Yannis; MAYFIELD, James; BRADSHAW, Jeff (ed.) KQML as an agent communication language in *: Software Agents*. MIT Press, Cambridge: 1997.

(Franklin 1996) FRANKLIN, S.; GRAESSER, A. Is It an Agent or Just a Program? A Taxonomy for Autonomous Agents. In *Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages*. New York : Springer-Verlag – 1996.

(Fridrich 2002) FRIDRICH, J., GOLJAN, M.: Practical steganalysis of digital images - state of the art. In: *Proceedings of the SPIE Photonics West (Security and Watermarking of Multimedia Contents IV).* Volume 4675. San Jose, California, USA – 2002.

(Fridrich 2004) FRIDRICH, Jessica; GOLJAN, Miroslav: On estimation of secret message length in LSB steganography in spatial domain. *Security, Steganography, and Watermarking of Multimedia Contents ,* 2004.

(Gonzalez 2002) GONZALEZ, Flight Lieutenant Fernando C.: *Counter Terrorist Steganography Search Engine*. Master Thesis, Department of Aerospace, Power and Sensors Cranfield University. Cranfield (UK), 2002.

(Google 2007) GOOGLE ©. Site de buscas. Endereço: http://www.google.com.br .

(Hair 1998) HAIR, Joseph F.; ANDERSON, Rolph E.; TATHAM, Ronald L.;

BLACK, William. *Multivariate Data Analysis*. 5th ed. Prentice Hall, Upper Saddle River, NJ, USA: 1998. ISBN 0-13-894858-5.

(Hussain 2005) HUSSAIN, Z. Metaheuristic Applications and Their Solutions Quality in *proceedings of Information and Communication Technologies, 2005. ICICT 2005.* First International Conference on, Vol., Iss., 27-28 Aug. 2005. Pages: 101- 104.

(Jonsson 2003) JONSSON, J.; KALISKI, B. 2003 *Public-Key Cryptography Standards (Pkcs) #1: RSA Cryptography Specifications Version 2.1*. RFC. ACM RFC Editor. 2003.

(Katzenbeisser 2000) KATZENBEISSER, Stefan; PETITCOLAS, Fabien A. P. *Information Hiding Techniques for Steganography and Digital Watermarking*. Boston: Artech House, 2000.

(Kipper 2004) Kipper, G. *Investigator's Guide to Steganography*. CRC Press. Boca Raton, Florida. 2004.

(Kofler 2003) KOFLER, R. KRIMMER, R. PROSSER, A.: *Electronic Voting: Algorithmic and implementation Issue*. System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on , 6-9 Jan. 2003.

(Lander 1997) LANDER, S. E.: Issues in Multiagent Design Systems. in *IEEE Expert*, vol 12, issue 2, p. 18-36. 1997.

(Latham 2006) LATHAM, Allan. *JPHS Steganography*. Disponível na Internet, http://linux01.gwdg.de/~alatham/stego.html. 24 ago. 2006.

(Luciano 2003) LUCIANO, E. M.; TESTA, M. G.; FREITAS, H. : *As tendências em comércio eletrônico com base em recentes congressos*. XXXVIII CLADEA, Lima/Peru, 2003.

(Lyu 2006) LYU, S.; FARID, H.: Steganalysis Using Higher-Order Image Statistics, *IEEE Transactions on Information Forensics and Security*, Vol. 1, No 1, March 2006.

(Macedo 2001) MACEDO, A. P. Cunha: *Metodologias de Negociação em Sistemas Multi-Agentes para Empresas Virtuais*. Tese de Doutoramento, Faculdade de Engenharia, Universidade do Porto, 2001.

(Macq 1995) MACQ, B. M.; QUISQUATER , J-J. *Cryptology for digital TV broadcasting*. Proceedings of the IEEE , 1995.

(Miche 2006) MICHE, Yoan; ROUE, Benoit; LENDASSE, Amaury; BAS, Patrick : *A Feature Selection Methodology for Steganalysis*. MRCS 2006, LNCS 4105, p. 49–56. Springer-Verlag Berlin Heidelberg, 2006.

(Panait 2005) Liviu Panait and Sean Luke. Cooperative Multi-Agent Learning: The State of the Art. *Autonomous Agents and Multi-Agent Systems*, 11(3):387–434, 2005.

(Patiño-Escarcina 2004) PATIÑO-ESCARCINA, R.E.; BEDREGAL, B.R.C. and LYRA, A. Interval Computing in Neural Networks: One Layer Interval Neural Networks in *Proceeding of 7th International Conference on Information Technology*, Hyderabad, India, December 20-23, 2004. In LNCS, V. 3356, Springer-Verlag, 2004.

(Paurobally 2002) PAUROBALLY, S.: *Rational Agents and the Processes and States of Negotiation*. Imperial College, Ph.D. Thesis, 2002.

(Pellegrini 2005) PELLEGRINI, J, BERTACCHI J. E. F and VITA, J.P.R -- *Forense Computacional*. Uma curta introdução à Forense Computacional em sistemas Unix. 2005.

(Phister 2004) PHISTER JR., Paul W.; PLONISCH, Igor G. Military Applications of Information Technologies. in *Air & Space Power Journal, Spring 2004*. Alabama, US.

(Quinlan 1986) QUINLAN, J. R. Induction of Decision Trees. *Machine Learning* 1, p. 81-106. Kluwer Academic Publishers, Boston, 1986.

(Rezende REZENDE, S.. (Org.). Conceitos sobre Aprendizado de Máquina. In:

2003a) *Sistemas inteligentes: fundamentos e aplicações*. 1ª ed. Barueri, São Paulo: Manole, 2003, cap 4.

(Rezende 2003b) REZENDE, S.. (Org.). Agentes e sistemas multiagentes. In: *Sistemas inteligentes: fundamentos e aplicações*. 1ª ed. Barueri, São Paulo: Manole, 2003, cap 11.

(Rocha 2006) ROCHA, Anderson de Rezende. *Randomização progressiva para esteganálise*. Dissertação (mestrado) - Universidade Estadual de Campinas, Instituto de Computação. SP : 2006.

(Russel 1995) RUSELL, Stuart J.; NORVING, Peter. Artificial Inteligence: A Modern Approach. Prentice-Hall Series in Artificial Inteligence, 1995.

(Sanches 2004) SANCHES, M. K.; GEROMINI, M. R.; *Aprendizado de Máquina*: Relatório Técnico. Instituto de Ciências Matemáticas e Computação, Universidade de São Paulo, 2004.

(Shaohui 2003) SHAOHUI, Liu; HONGXUN, Yao; WEN, Gao: Neural Network based Steganalysis in Still Images. *Proceedings of IEEE ICME 2003*.

(Singh 2001) SINGH, Simon. *O Livro dos Códigos - A Ciência do Sigilo: do Antigo Egito à Criptografia Quântica*. Ed. Record, Rio de Janeiro, 2001.

(Siqueira 2005) SIQUEIRA, Paulo Henrique: *Uma nova abordagem na resolução do problema do Caixeiro Viajante*. Tese de Doutorado, Programa de Pós-Graduação em Métodos Numéricos em Engenharia, Universidade Federal do Paraná. 2005

(Sison 1998) SISON, R.; MASSAMICHI, S.: Student Modeling and Machine Learning in *International Journal of Artificial Intelligence in Education*, vol. 9, pag. 128-158, 1998.

(Sun 2007) SUN Microsystems Inc. *Java Technology*. Disponível na Internet: http://java.sun.com/ . 29 jun 2007.

(Sung 2004) SUNG, A. H., TADIPARTHI, G. R., MUKKAMALA, S. Defeating the Current Steganalysis Techniques (Robust Steganography). In

Proceedings of the international Conference on information Technology: Coding and Computing (Itcc'04) Volume 2 - Volume 2 (April 05 - 07, 2004). ITCC. IEEE Computer Society, Washington, DC, 440.

(Tzchoppe 2003) TZSCHOPPE, Roman; BÄUML, Robert; HUBER, Johannes B., KAUP, Andre: Steganographic System Based on Higher-Order Statistics. *Proc. of SPIE Vol. 5020, Security and Watermarking of Multimedia Contents V*, Santa Clara, California, USA, 2003.

(Wang 2004) WANG, H.; WANG, S.: Cyber warfare: steganography vs. steganalysis. *in Communications of ACM*, vol. 47, 10 (Oct. 2004), 76-82.

(Wang 2007) WANG, Y; MOULIN, P: *Optimized feature extraction for learning-based image steganalysis.* IEEE Trans. Inform. Forensics and Security, vol. 2, no. 1, Mar. 2007.

(Weka 2007) Weka 3: *Data Mining with Open Source Machine Learning Software in Java*. Universidade de Waikato. Nova Zelândia. Site da Web: http://www.cs.waikato.ac.nz/ml/weka/ . Última visita 5 jul 2007.

(Wooldridge 2001) WOOLRIDGE, Michael J., *Introduction to Multiagent Systems*, John Wiley & Sons, Inc., New York, NY, 2001.

(Zhang 2006) ZHANG, H.; BERG, A. C.; MAIRE, M. MALIK, J.: *SVM-KNN:* Discriminative Nearest Neighbor Classification for Visual Category Recognition *in Computer Vision and Pattern Recognition.* v. *2*, p. 2126-2136, 2006.

(Zimmermann 1995) ZIMMERMANN, P. *The Official PGP User's Guide.* MIT Press, 1995, ISBN: 0-262-74017-6, 216 p.

Anexo A : Comparativo

A tabela a seguir apresenta um comparativo das características dos trabalhos relacionados e da nossa abordagem.

Abordagem de Esteganálise	**Tipo**	**Tamanho da Base**	**Descrição**	**Técnicas Detectadas**	**Método de Aprendizado Utilizado na Classificação**	**Tipos de Imagens**	**Resultado**	**Taxa de Acerto (%)**
Lyu (2006)	Universal	40.000 imagens.	Análise de dados estatísticos de primeira e alta ordem LSB	Jsteg, Outguess, Steghide, Jphide, e F5.	SVM.	Coloridas, de interior ou exterior, no formato JPEG.	Detecta presença de dados ocultos	20% para mensagens ocultas com 10% do tamanho da cobertura, e 99% para mensagens ocultas com 50% do tamanho da cobertura
Fridrich (2004)	Específica	60 imagens.	Análise Estatística dos bits	J-Steg e JPHide/ JPSeek.		Em escala de cinza, tiradas de um câmera digital	Estima o tamanho de dados ocultos.	Não apresenta taxa de acerto.

			menos significativos.			Cannon G2.		
Shaohui (2003)	Específica	88 imagens.	Busca no domínio das trasformadas (DCT, DFT e DWT)	*Quantization index modulation*.	Redes Neurais	Não especifica.	Detecta presença de dados ocultos	80%
Rocha(2006)	Específica	20.000 imagens.	Análise estatística LSB progressiva	?	Arvores de Decisão, SVM, LDA, com e sem Agrupamento (Bagging)	Formato PNG, arte, interior e exterior.	Detecta presença de dados ocultos	De 70,9% a 80,2% (ind.) e de 78% (*bagging*).
Nossa Abordagem	Específico a Universal*	300 imagens.	SMA multi-classificador, a partir de dados estatisticos	JPHide/ JPSeek.	Arvores de decisão e outros**	Coloridas:Paisagens, interiores, animais, construções, pessoas, plantas e alimentos.	Detectar presença de dados ocultos ***	82% para mensagens de 10% do tamanho da cobertura .

Tabela A. Comparativo dos métodos de esteganálise citados com o método apresentado.

*específica, capaz de atingir a universalidade através da adição de novos agentes, utilizando-se da filosofia do "dividir para conquistar";

** com perspectivas de adicionar outros;

***perspectivas futuras de extração de dados ocultos;**

Livros Grátis

(http://www.livrosgratis.com.br)

Milhares de Livros para Download:

Baixar livros de Literatura
Baixar livros de Literatura de Cordel
Baixar livros de Literatura Infantil
Baixar livros de Matemática
Baixar livros de Medicina
Baixar livros de Medicina Veterinária
Baixar livros de Meio Ambiente
Baixar livros de Meteorologia
Baixar Monografias e TCC
Baixar livros Multidisciplinar
Baixar livros de Música
Baixar livros de Psicologia
Baixar livros de Química
Baixar livros de Saúde Coletiva
Baixar livros de Serviço Social
Baixar livros de Sociologia
Baixar livros de Teologia
Baixar livros de Trabalho
Baixar livros de Turismo