UNIVERSIDADE FEDERAL DO CEARÁ CAMPUS QUIXADÁ
Bacharelado em Sistemas de Informação

KAYNAN COELHO LIRA

AUXILIANDO O DESEMPENHO DE ALUNOS COM TENDÊNCIA A EVASÃO NA EDUCAÇÃO A DISTÂNCIA UTILIZANDO TÉCNICAS DE MINERAÇÃO DE DADOS E SISTEMAS MULTIAGENTES

Quixadá-CE
2016

KAYNAN COELHO LIRA

AUXILIANDO O DESEMPENHO DE ALUNOS COM TENDÊNCIA A EVASÃO NA EDUCAÇÃO A DISTÂNCIA UTILIZANDO TÉCNICAS DE MINERAÇÃO DE DADOS E SISTEMAS MULTIAGENTES

Monografia apresentada ao curso de Sistemas de Informação da Universidade Federal do Ceará Campus Quixadá, como requisito para obtenção do título de Bacharel em Sistemas de Informação.

Orientador: Dr. Marcos Antônio de Oliveira

Quixadá-CE
2016

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

C617a Coelho Lira, Kaynan.
Auxiliando o desempenho de alunos com tendência a evasão na educação a distância utilizando técnicas de mineração de dados e sistemas multiagentes / Kaynan Coelho Lira. – 2016.
51 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de Sistemas de Informação, Quixadá, 2016.
Orientação: Prof. Dr. Marcos Antônio de Oliveira.

1. Mineração de Dados. 2. Ensino à Distância. 3. Agentes inteligentes (Software). 4. Evasão Escolar. I. Título.

CDD 005

KAYNAN COELHO LIRA

AUXILIANDO O DESEMPENHO DE ALUNOS COM TENDÊNCIA A EVASÃO NA EDUCAÇÃO A DISTÂNCIA UTILIZANDO TÉCNICAS DE MINERAÇÃO DE DADOS E SISTEMAS MULTIAGENTES

Monografia apresentada ao curso de Sistemas de Informação da Universidade Federal do Ceará Campus Quixadá, como requisito para obtenção do título de Bacharel em Sistemas de Informação.

Aprovado em: _____/julho/2016

BANCA EXAMINADORA

_____

Dr. Marcos Antônio de Oliveira

_____

Prof. Regis Pires Magalhães

_____

Prof. Enyo José Tavares Gonçalves

À Maria de Fátima Lira, minha mãe e fonte de motivação.

À Antônia Clea Lira, minha avô e segunda mãe.

Aos amigos e demais familiares que torceram e ajudaram de alguma forma.

Ás crianças que carregarão o futuro da nossa existência.

RESUMO

A Educação a Distância (EAD) é uma modalidade de ensino que tem como principal ferramenta o Ambiente Virtual de Aprendizagem (AVA), onde professores, alunos e tutores podem interagir de forma direta ou indireta. Um problema comum na EAD é a elevada quantidade de alunos que acabam desistindo das aulas, porém é preciso acompanhar esses alunos periodicamente para alterar esse cenário, levando-os para o êxito em seus estudos. Com o grande aumento na quantidade de informações geradas pelos alunos interagindo nos AVA's, surgiu a oportunidade de utilizar Mineração de Dados para descobrir padrões nos comportamentos dos alunos, podendo identificar precocemente alunos que tendem a ter um bom ou mau desempenho. Esse trabalho descreve o desenvolvimento de um modulo para um Sistema Multiagente (SMA) desenvolvido pelo Grupo de Estudo em Engenharia de Software e Sistemas Multiagente (GESMA), tendo como finalidade acompanhar o comportamento dos alunos e identificar precocemente quando eles podem tender a evasão, acionando os demais agentes que compõem o sistema para auxiliarem o aluno em seu êxito escolar.

Palavras-chave: Mineração de Dados. Ensino à distância. Agentes inteligentes (Software). Evasão Escolar.

ABSTRACT

The distance education is a form of education that has as main tool the Virtual Learning Environment (VLE), where teachers, students and tutors can interact directly or indirectly. A common problem in EAD is the large amount of students who end up quitting school. However it is necessary to follow those students periodically to change this scenario and leading them to success in their studies. With the great increase in the amount of information generated by students interacting in VLE's, data mining comes as an opportunity to discover patterns on the behaviors of students, and with to be able to identify early students who tend to have a good or bad academic performance. This paper describes the development of a Multi-agent system module (MAS) developed by the Study Group in Software Engineering and Multi-agent Systems (GESMA) with the purpose to monitor the behavior of the students and identify as soon as possible when they tend to escape, engaging the other agents in the system to aid the student in their school success.

Keywords: Data mining. Distance learning. Intelligent agents (Software). School Dropout.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

A Educação a Distância (EAD) é uma modalidade de ensino que tem como principal ferramenta o Ambiente Virtual de Aprendizagem (AVA), onde professores, alunos e tutores podem interagir de forma direta ou indireta. Com grande flexibilidade de tempo e compromisso com essa modalidade de ensino, o discente necessita manter o foco, contato com os tutores e organização em seus horários para estudo, porém, não é possível desconsiderar a vida pessoal do indivíduo e é devido a essa flexibilidade que a EAD permite que apareçam alguns problemas em relação ao desempenho do aluno no decorrer do curso. Muitos alunos chegam a desistir/evadir por causa de problemas financeiros, falta de tempo para o comprometimento com os estudos, falta de material didático auxiliar disponível no AVA, falta de profissionalismo dos tutores, e entre outros fatores (CAVALCANTI et al., 2014).

Devido ao aumento de alunos na EAD, gerenciar seus processos de aprendizagem com qualidade de interação e de acompanhamento dentro de um AVA, visando o êxito e a permanência dos alunos nos seus respectivos cursos, é uma tarefa que exige cada vez mais dos professores. Os dados gerados nas interações entre professores e alunos, dos alunos entre si e deles com os recursos disponibilizados no AVA, são volumosos e pouco explorados, podendo conter informações uteis para a instituição, porém reuni-los e interpreta-los é uma atividade complexa e exaustiva (KAMPFF, 2009).

Acompanhar o aluno no decorrer do curso é fundamental para o êxito no curso. Com um bom sistema de acompanhamento e avaliação, é possível observar características que representam suas dificuldades e assim poderia ser oferecido o tipo adequado de ajuda. Existem algumas soluções desenvolvidas para auxiliar o acompanhamento do desempenho de alunos em um AVA, como por exemplo o Sistema Multiagente desenvolvido pelo Grupo de Estudo de Engenharia de Software em Sistemas Multiagente (GESMA) da Universidade Federal do Ceará (UFC) Campus Quixadá que integra a Universidade Aberta do Brasil (UAB) da Universidade Estadual do Ceará (UECE), denominado *SMA Moodle* (GONÇALVES et al., 2014). Este sistema tem como principal objetivo auxiliar o acompanhamento de alunos no AVA *Moodle*[1], plataforma de Educação a Distância utilizada mundialmente e pela Universidade Estadual do Ceará (UECE), que utiliza a modalidade semi presencial, cujo os alunos cujo em alguns casos os alunos realizam atividades presencias. O sistema é composto por um conjunto de agentes,

[1]Disponível em: https://moodle.org

onde cada agente através de seus compromissos, se responsabiliza por uma parte do AVA. Algumas das funcionalidades desse SMA são: acompanhar o desempenho do aluno durante os cursos matriculados, acompanhar as atividades dos tutores dos respectivos cursos, criar grupos de alunos de acordo com o perfil e temas de interesse e enviar materiais de apoio aos alunos e tutores.

Este trabalho teve como principal objetivo o desenvolvimento de um módulo que foi integrado ao SMA. Este modulo será responsável pela identificação prévia de características que representam comportamentos que podem levar o aluno a evadir ou a ter mau desempenho no curso. Para que isso fosse possível, foram analisados dados históricos dos alunos da UECE. Esses dados sofreram um processo de clusterização para dividir em grupos os perfis dos alunos, e através de classificação foi possível prever o desempenho dos alunos para que fossem ajudados pelo SMA. Com esses valores de desempenho identificados, informações são repassadas aos demais agentes do sistema para que decisões sejam tomadas em conjunto por eles. Antecipar a identificação desses perfis é de grande utilidade e interesse das instituições de ensino que têm como método de ensino a EAD, pois tanto os docentes poderão remediar da melhor forma a situação, como os discentes terão um acompanhamento mais adequado no decorrer do curso, e, por sua vez, diminuindo a quantidade de alunos que podem evadir, resultando no aumento de concludentes dos cursos.

O intuito desse projeto é propor uma abordagem que irá somar com o SMA desenvolvido pelo grupo GESMA, aumentando sua utilidade e funcionalidades para o corpo docente e discente que venham a utilizar o AVA *Moodle*.

Esse trabalho está divido nas seguintes seções: a Seção 2 descreve os principais conceitos usados no trabalho; a Seção 3 apresenta alguns trabalhos relacionados; a Seção 4 descreve o experimento e os resultados, bem como a análise dos mesmos; e a Seção 5 conclui e pincela sobre os trabalhos futuros para tornar o modulo mais robusto e preciso para o acompanhamento dos alunos.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo tem como princípio apresentar os principais conceitos chave para a compreensão deste trabalho. Os conceitos que serão apresentados são, **Evasão na Educação a Distância**, **Mineração de Dados Educacionais**, **Extração de Conhecimento** e **Sistemas Multiagente**. Os conceitos destacados são os elementos mais importantes que proporcionam a base para o desenvolvimento deste trabalho.

2.1 Evasão na Educação a Distância

A Educação a Distância (EAD) é o nome atribuído a uma modalidade de ensino que tem como característica principal o ensino e aprendizagem em que alunos e professores não necessitam estar juntos em um ambiente físico como uma sala de aula durante a maior parte do tempo do curso (CAMBRUZZI, 2014). A interação entre docentes e discentes acontece através de Ambientes Virtuais de Aprendizagem (AVA), o *Moodle* é um exemplo de AVA, onde professores, tutores e alunos podem interagir entre si. Dependendo das políticas e normas de ensino da instituição, alguns encontros presenciais podem acontecer periodicamente.

Um dos benefícios proporcionados pela EAD é a capacidade de ampliar oportunidades educativas aos indivíduos, desconsiderando qualquer limitação geográfica ou socioeconômica. Devido a essa capacidade de conseguir levar a educação a usuários de diversos lugares, como também a flexibilidade de horários e compromissos com as atividades dos cursos, a EAD é estimulada tanto por iniciativa privada como pública.

Muitos alunos encontram dificuldades em se adaptarem a EAD, isso acontece por causa do método padrão de ensino presencial. Algumas das dificuldades mais encontradas são a falta de tempo e organização dos horários de estudo, a dificuldade de se adaptar a uma tecnologia nova e uma nova forma de aprendizagem onde o aluno tem que se disciplinar e manter o foco. Devido a essas dificuldades encontradas, nos deparamos com um problema na EAD, que é o elevado índice de evasão dos alunos em relação aos seus cursos (KAMPFF, 2009).

Podemos entender por Evasão na Educação a Distância o ato do aluno desistir do curso em que esteja devidamente matriculado antes da sua conclusão (CAMBRUZZI, 2014). Dentre os motivos que podem levar ao aluno desistir do curso ou a ter características que podem

influencia-lo a ter um mau desempenho no curso, podemos dividi-los em duas categorias: Causas Internas e Causas Externas (KAMPFF, 2009).

Podemos entender como Causas Internas os fatores que partem do AVA ao aluno, como a falta de adaptação à tecnologia utilizada para a aprendizagem, aluno não ter contato constante com computadores, a falta de compromisso e qualificação dos tutores, a falta de disponibilização de materiais de estudo mais didáticos e em alguns casos, o AVA não segue padrões de usabilidade deixando a desejar a facilidade de interação do usuário com a plataforma.

Como Causas Externas podemos entender como os fatores que partem do aluno com a sua vida pessoal, como a rotina corrida do dia a dia o impedindo de ter um horário reservado para se dedicar aos estudos, problemas financeiros, problemas emocionais e por não praticar a auto-disciplina.

Para uma instituição de ensino é fundamental identificar os alunos e as causas que podem leva-los a evasão e assim encontrar formas de lidar com essa situação, proporcionando um melhor acompanhamento ao aluno durante o seu percurso do início à conclusão do curso.

Em um AVA dados são depositados pelos alunos constantemente. Esses dados podem ser compreendidos como a forma que o aluno se comporta no AVA. E através da Mineração de Dados é possível transforma-los em informações. Entre essas informações é possível identificar características que podem apontar perfis de alunos que possuem mau desempenho, que tendem à evasão, bom desempenho e entre outros.

2.2 Mineração de Dados Educacionais

Devido ao rápido avanço das tecnologias de coleta e armazenamento de dados, as organizações passaram a acumular uma vasta quantidade de dados (TAN et al., 2006). Devido a isso foi possível perceber que através deles poderiam ser obtidas informações úteis através da MD (Mineração de Dados), que é uma tecnologia composta por métodos tradicionais de análise de dados com novos algoritmos sofisticados para processar essa quantidade de dados.

Com o grande acúmulo de dados em Instituições de Ensino, surgiu uma subárea da Mineração de Dados, a Mineração de Dados Educacionais (MDE). Esta é uma área em expansão, tendo como principais enfoques os trabalhos relacionados com aprendizagem supervisionada, que é quando o algoritmo é treinado usando exemplos rotulados como uma entrada

onde a saída desejada é conhecida, e aprendizagem não supervisionada, que é quando o algoritmo desconhece as classes que rotulam os dados históricos, sendo necessário o próprio algoritmo encontrar e reconhecer os padrões (CAMBRUZZI, 2014).

Essas duas formas de aprendizagem se dividem em técnicas utilizadas no processo de MD. Algumas delas são:

1. Aprendizagem Supervisionada

    - Classificação: é utilizada para identificar modelos ou subgrupos de dados classificados de acordo com variáveis previamente definidas.
    - Regressão: é uma atividade utilizada para prever valores de dados de acordo com uma função de mapeamento obtida.

2. Aprendizagem Não Supervisionada:

    - Associação: é a atividade responsável por encontrar grupos de dados que possuem relação características em comum.
    - Clusterização: é utilizado para identificar conjunto de categorias ou agrupamentos que possam descrever o comportamento dos dados selecionados.

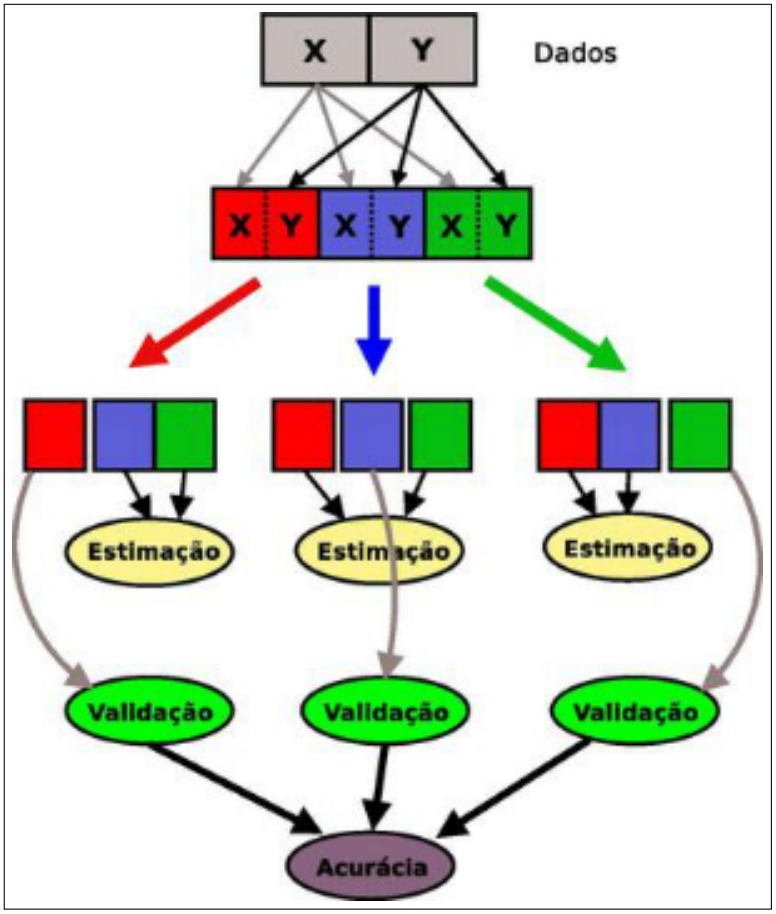
Para cada técnica existe um conjunto de algoritmos. Em um processo de MD é possível selecionar um ou mais algoritmos que sejam necessários para que seja possível obter o resultado esperado.

Um algoritmo de mineração de dados nada mais é que um conjunto de cálculos e heurísticas que a partir de um modelo, eles identificam padrões ou são capazes de predizer valores de atributos. Para que seja possível criar um modelo, o algoritmo analisa os dados fornecidos, verifica padrões e tendências. A escolha do algoritmo adequado não é fácil, pois é necessário analisar o problema do domínio em questão, o conjunto de dados e seus respectivos tipos, o tamanho da base de dados, e entre outros fatores.

Tão importante como definir o modelo de dados encontrado através da MD, é a sua análise e validação. No caso do presente trabalho, que tem como finalidade utilizar predição, foi utilizada a técnica de *cross-validation* através do método *k-fold*. Essa técnica tem como finalidade dividir o conjunto total de dados em k conjuntos iguais. Após o processo de particionamento, um subconjunto é utilizado para teste, os demais k-1 subconjuntos são utilizados para estimar os parâmetros e calcular a acurácia do modelo. Esse método irá repetir k vezes

esse processo, alternando circularmente o subconjunto de testes (HAN; PEI; KAMBER, 2011). A figura 1 explica visualmente em um exemplo como acontece esse processo.

Figura 1 – *Cross-Validation (k-fold)*



Fonte: capturada do Wikipédia

Para o presente trabalho de acordo com os trabalhos relacionados e pesquisas relacionadas, os algoritmos que mais se adequam para a solução do problema em questão, são algoritmos de classificação, por serem mais eficientes para prever ou descrever conjunto de dados de acordo com categorias nominais (evadido, aprovado, reprovado). Dentre eles os que mais se destacam são:

1. ***RuleLearner***: algoritmo utilizado em técnicas de classificação, que funciona de forma similar ao algoritmo *Repeated Incremental Pruning to Produce Error Reduction* (RIPPER) que é um algoritmo de classificação de eventos que utiliza uma coleção de regras no formato (Se ***condição*** Então ***classificação***), para geração das regras ele tem como critério a *accuracy* (precisão) (COHEN, 1995).

2. ***ADTree***: algoritmo de classificação por árvore de decisão, conhecido também como algoritmo de árvore de decisão alternada (FREUND; MASON, 1999).

3. ***SimpleCart***: é um algoritmo derivado da implementação do algoritmo *Classification and Regression Trees* (CART) que é uma árvore de decisão binária que é construída pela divisão de um nó em dois nós filhos repetidamente. Ela começa com o nó raiz que contém toda a amostra de aprendizagem (BREIMAN et al., 1984).

4. ***J48***: algoritmo utilizado em técnicas de classificação por árvore de decisão. Com essa técnica uma árvore de regras é construída para modelar o processo de classificação. Após a sua criação ela é aplicada a cada tupla do banco de dados para obter os resultados (QUINLAN, 2014).

5. ***Random Forest***: é um classificador composto por uma coleção de árvores $\{h_k(x)\}$,k = 1,2, ...,L, onde T_k é um conjunto de amostras aleatórias independentes e identicamente distribuídas, no qual cada árvore vota na classe mais popular para a entrada x (BREIMAN, 2001).

Problemas como evasão de alunos ou mau desempenho em um curso a distância podem ser identificados previamente por técnicas de MD. Gerar esses diagnósticos e identificar os perfis de alunos com essas características é de grande importância para a instituição, pois assim novas formas de resolver esses problemas podem ser desenvolvidas, proporcionando um melhor acompanhamento ao discente.

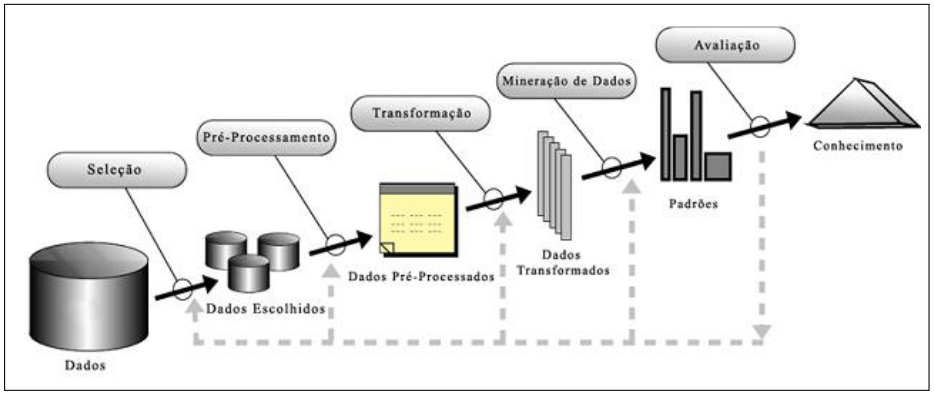
Tendo como objetivo obter informações úteis à instituição através da MDE, é necessário o desenvolvimento de uma abordagem mais ampla que faça um estudo prévio dos fatores a serem monitorados. Tão importante quanto a seleção dos algoritmos e dos atributos a serem analisados, é a forma que essas informações serão obtidas e de que forma essa analise irá se adaptar periodicamente a quantidade de eventos que podem acontecer.

2.3 Extração de Conhecimento (KDD – Knowledge Discovery in Databases)

Extração do conhecimento ou processo de KDD derivado do inglês *knowledge- discovery in databases*, é um processo que busca identificar potenciais padrões úteis, que estejam

embutidos nos dados e, tornando-os compreensíveis para um determinado contexto (FAYYAD et al., 1996). A figura 2 demonstra sequencialmente as etapas para descoberta de conhecimento.

Figura 2 – Processo de KDD



Fonte: (FAYYAD et al., 1996)

O processo de KDD consiste em uma sequência de etapas que devem ser executadas sequencialmente, pois ao final de cada etapa, o resultado obtido serve de auxílio para a etapa seguinte, podendo repetir etapas anteriores sempre que necessário. São etapas do processo de KDD: seleção, pré-processamento e limpeza, transformação dos dados, mineração de dados,interpretação e avaliação dos dados. A seguir cada subseção irá explorar sucintamente cada uma das etapas do processo de KDD.

2.3.1 Seleção

A etapa de Seleção é a primeira no processo de KDD, é a etapa em que serão definidas a (s) fonte (s) que se relacionam com o domínio para a extração dos dados apropriados para o contexto. Ela é trivial para obter êxito no resultado final. Devido as fontes dos dados poderem vir de fontes heterogêneas (planilhas, banco de dados relacional/não relacional, formulários) como também possuírem diversos formatos (CSV, ARFF, TXT), ela se torna uma atividade complexa, assim sendo necessário padroniza-los, integra-los e limpa-los.

2.3.2 Pré-processamento e Limpeza

Devido os dados estarem possivelmente em formatos diferentes como também terem sido coletados de fontes diferentes, o subconjunto selecionado pode vir com alguns erros, como dados ausentes, dados com erro, registros duplicados e ruídos, tornando-se necessário tratar esses dados por um processo de integração, padronização e limpeza, para que seja gerado um subconjunto de dados que possa representar o domínio.

Ao realizar as operações dessa etapa, apesar de existirem ferramentas que podem auxiliar nesse processo, é de grande importância a presença de um especialista do domínio. Ele é quem mais entende a situação abordada e quem está mais apto a selecionar quais dados são relevantes ou que necessitam ser removidos.

De acordo com Oliveira (2000) essa etapa de pré-processamento e limpeza compreende os seguintes aspectos:

1. **Padronização dos Valores dos Atributos**: devido ao fato de que os dados possam ser recuperados de várias fontes diferentes, é possível que dados que representam atributos com o mesmo significado possuam tipos diferentes. Por exemplo, o controle de acesso de funcionários em uma determinada base de dados está representada por “DIRETOR”, “SUPERVISOR”, “ATENDENTE”, enquanto em outra base de dados está representado por “1”, “2”, “3”, portanto, é necessário padronizar esses dados para um tipo em comum.

2. **Remoção de Registros Duplicados**: em determinadas situações após o processo de integração dos dados, podem aparecer registros duplicados ou que representem a mesma informação, porém de forma diferente.

3. **Tratamento e Eliminação de Ruídos**: durante o processo de coleta dos dados desejados é possível que alguns valores possam conter ruídos devido a alguma falha no processo de coleta, por exemplo, tipos de dados não suportados pelo SGBD. Os dados com ruídos em seus valores devem ser corrigidos atribuindo a eles os seus valores corretos ou eliminados de acordo com a relevância do mesmo para o processo de KDD.

4. **Tratamento de Valores Ausentes**: encontrar campos em tabelas de banco de dados nulos ou formulários com campos ignorados e não preenchidos é tão comum quanto encontrar dados com valores duplicados. Para solucionar esse problema é preciso estabelecer re-

gras e critérios para correção para decidir se esses dados irão ser ignorados, preenchidos com o valor correspondente ou algum padrão para valoração de campos nulos, e sempre procurando resolver com o método mais adequado que influencia positivamente para o processo de KDD.

Após a realização dessa etapa, os dados já estarão selecionados, pré-processados e limpos, porém ainda não foram formatados adequadamente para que os algoritmos da etapa de MD sejam aplicados.

2.3.3 Transformação dos Dados

Esta etapa é crucial para que os algoritmos que serão aplicados na etapa de MD consigam obter êxito em seus objetivos com eficiência e eficácia, portanto, faz-se necessário realizar a formatação e o armazenamento adequado dos dados.

Durante esta etapa é possível obter valores através de outros dados, denominando-os de valores derivados. Por exemplo o lucro mensal de uma empresa que poderá ser obtido através da somatória das transações efetuadas durante o mês desejado.

2.3.4 Mineração dos Dados

Esta etapa é a mais importante de todo o processo de KDD, pois é nela onde as informações relevantes são obtidas.

De acordo Tan (2006), MD é uma forma de explorar e analisar dados de forma supervisionada ou não supervisionada, com o intuito de perceber regras e padrões em grandes fontes de dados afim de obter informações relevantes para algum objetivo ou entidade. Mais detalhes sobre MD já foram descritos na Seção 2.2.

2.3.5 Interpretação e Avaliação

Através dessa etapa é possível chegar à informação desejada, através da interpretação e avaliação dos padrões encontrados através da etapa de MD. Os usuários podem utilizar diversas ferramentas com funcionalidades estatísticas e de visualização para validarem ou julgarem um padrão irrelevante.

Ao finalizar essa etapa, caso não sejam encontradas informações relevantes ou informações esperadas, é preciso retornar aos passos anteriores para corrigir os possíveis problemas até encontrar as informações necessárias.

De acordo com a literatura da área, existem várias ferramentas que podem auxiliar durante o processo de KDD. Dentre elas é possível citar o RapidMiner[1], Weka[2] e o Elki[3], que são ferramentas de código aberto desenvolvidas em Java.

2.4 Sistemas Multiagentes

Sistemas Multiagentes podem ser entendidos como uma subárea da inteligência artificial, composta por agentes que, segundo Russel e Norvig (2004), são entidades de software capaz de perceber seu ambiente por meio de sensores e de agir sobre ambientes por intermédio de atuadores, podendo comunicar-se e tendo como princípio conquistar seus objetivos firmados em seus respectivos compromissos.

De acordo com a literatura da área é possível encontrar *frameworks* destinados ao desenvolvimento de SMA's. Dentre estes é possível destacar a plataforma JADE[4](*Java Agent Development Enterprise*). Este *framework* foi desenvolvido na linguagem Java, além de ser um ambiente de execução de agentes, ele simplifica o desenvolvimento de SMA's através de uma arquitetura que está de acordo com as especificações FIPA[5] (*Foundations of Intelligent Physical Agents*).

A FIPA é uma organização formada para produzir especificações de padrões de soft-

[1]Disponível em: https://rapidminer.com
[2]Disponível em: http://www.cs.waikato.ac.nz/ml/weka
[3]Disponível em: http://elki.dbs.ifi.lmu.de
[4]Disponível em: http://jade.tilab.com
[5]Disponível em: http://www.fipa.org

ware para agentes. Suas especificações consistem em representar um conjunto de normas para promover interação de agentes e os seus serviços. Como exemplo de um possível SMA, será descrita a situação a seguir, dando ênfase aos agentes e à comunicação entre eles.

Em uma determinada cidade existe um corpo de bombeiros, ele possui bombeiros que podem ser entendidos como agentes que possuem o papel de apagar incêndios. Na mesma cidade existem pessoas que como agentes têm o papel de avisar acontecimentos de incêndios. Aqui identificamos dois agentes, o Agente Bombeiro e o Agente Alarmador. Quando acontecer algum incêndio o Agente Alarmador irá enviar uma mensagem para o Agente Bombeiro, por sua vez o Agente Bombeiro irá iniciar os procedimentos para combater o incêndio (BATISTA, 2008).

Nessa situação os dois agentes descritos possuem papéis e características diferentes. Aqui é possível identificar uma aridade binária de acordo com o protocolo de comunicação das especificações FIPA. Uma aridade binaria representa a comunicação de um único emissor com um único receptor. De acordo com as subdivisões dos protocolos de comunicação, também existem casos que os agentes possuem aridade n, isso implica que existe um emissor e vários receptores. De acordo com as especificações FIPA, um protocolo de comunicação entre agentes possui a seguinte estrutura de dados, emissor, receptor (es), linguagem utilizada, funções de codificação e decodificação da linguagem e ações que o receptor deve executar.

-

3 TRABALHOS RELACIONADOS

Neste capítulo serão apresentados trabalhos que se relacionam com o tema abordado. O objetivo foi identificar métodos já existentes para o problema em questão e analisar os seus benefícios comparando-os com a proposta desta pesquisa. Ao final do capítulo será apresentada uma tabela comparando as principais características entre os trabalhos aqui citados com o presente trabalho.

3.1 Mineração de Dados Educacionais para Geração de Alertas em Ambientes Virtuais de Aprendizagem como Apoio à Pratica Docente

Kampff (2009) tem como principal objetivo propor uma arquitetura para sistemas de alertas em AVA. Essa arquitetura será baseada em informações extraídas por processos de Mineração de Dados, buscando identificar alunos com características e comportamentos que podem levar à evasão ou à reprovação.

Ele apresenta como problema que, a grande quantidade de alunos por turma na Educação a Distância como também as qualificações de professores, principalmente no ensino superior, não possuem preparação adequada para a pratica docente e em muitos casos não possuem formação pedagógica. Outro fator é a falta de experiência em dominar ferramentas que auxiliem a mediação das atividades em EAD. Portanto, tem se tornado cada vez mais difícil gerenciar e acompanhar o desempenho desses alunos sem a utilização de uma ferramenta auxiliar.

Kampff (2009) justifica sua proposta apresentando as seguintes hipóteses:

1. Através da Mineração de Dados Educacionais será possível identificar características e comportamentos dos alunos que podem ser uteis para a pratica docente.

2. A geração de alertas, tendo como base as informações obtidas no processo de Mineração de Dados, servirá para alertar o corpo docente da instituição sobre possíveis alunos com tendência à evasão ou reprovação, para que medidas preventivas sejam aplicadas.

De acordo com Kampff (2009), os fatores responsáveis que podem levar o aluno a evadir do curso, ou obter reprovação no mesmo, podem ser divididos em duas categorias: fatores internos a instituição e fatores externos, conforme descritos na Seção 2.1. Essa categorização

será utilizada pelo autor como base de orientação para estudos relacionados aos possíveis fatores e causas para evasão e reprovação na EAD.

Para a etapa de experimentação do seu sistema de alertas, Kampff (2009) utilizou o AVA NetAula utilizado pela Universidade Luterana do Brasil (ULBRA). A base de dados do AVA foi analisada detalhadamente para que fosse possível a seleção dos atributos mais relevantes e dos algoritmos mais adequados para o processo de extração de conhecimento. Ao final do processo de análise, pré-processamento, agrupamento e validação dos dados, foram totalizados 230 atributos para representar cada aluno, porém, apenas 87 destes atributos foram selecionados para a etapa de MD, dos quais os mais relevantes contemplam as seguintes categorias:

1. Demográficos: informações pessoais do aluno.
2. Comportamentais: informações relacionadas ao comportamento do aluno no AVA.
3. Desempenho: informações sobre entregas e notas das tarefas.
4. Desempenho final: relação do aluno com o resultado final de cada curso.

Para a etapa de MD, Kampff (2009) utilizou dois algoritmos (*DecisionTree* e o *RuleLearner*) já descritos na Seção 2.2, e a ferramenta RapidMiner, já descrita na Seção 2.3. O sistema de alertas desenvolvido por Kampff (2009) funciona através de geração de alertas definidos pelo professor (alertas fixos), e por alertas derivados da etapa de MD (alertas baseados em padrões).

Para validação dos dados obtidos, Kampff (2009) aplicou testes de hipóteses baseados nos percentuais de aprovação, evasão e reprovação dos alunos acompanhados, tendo como base comparativa os dados históricos dos alunos que não foram acompanhados pelo sistema de alertas. Pretende-se neste trabalho avaliar o desempenho do sistema nos dados históricos da UAB/UECE, verificando-se a existência ou não de tendência a evasão em alunos que já cursaram a UAB/UECE em algum momento no passado.

Através desses dados históricos, foi possível montar um modelo preditivo de classificação, que por sua vez, irá classificar novos alunos de acordo com seus respectivos dados históricos.

Apesar das semelhanças e das influencias, este trabalho se diferencia em alguns pontos com o trabalho apresentado por Kampff (2009). O presente trabalho não teve como finalidade desenvolver um sistema de alertas, mas se beneficiou das métricas utilizadas por Kampff (2009) para a criação do seu modelo preditivo.

3.2 Uma Abordagem Genérica de Identificação Precoce de Estudantes com Risco de Evasão em um AVA utilizando Técnicas de Mineração de Dados

O trabalho descrito em Cavalcanti (2014), tem como principal objetivo o desenvolvimento de uma abordagem genérica de identificação de tendência à evasão em cursos a distância que fazem uso de Ambientes Virtuais de Aprendizagem, aplicando técnicas de Mineração de Dados.

Em Cavalcanti (2014) é apresentada uma abordagem genérica para a identificação precoce de alunos que possam ter perfis que os levem a evasão, isso se torna possível através da utilização técnicas de Mineração de Dados. O foco de sua pesquisa não é somente uma única disciplina em um único período de tempo, mas sim a identificação de perfis de alunos nos mais diversos contextos de um AVA, abrangendo todos os cursos e em todos os períodos letivos.

Cavalcanti (2014) focou-se apenas em dados variantes no tempo para sua escolha. Ele justifica isto pelo fato de que dados variantes no tempo podem ser obtidos através do monitoramento dos alunos que utilizam um AVA. Estes dados não necessitam da elaboração de questionários para que sejam obtidos, tornando menos trabalhosa a etapa de pré-processamento e transformação dos dados, como também, é através desse tipo de dado que é possível prover modelos genéricos para o processo preditivo, pelo fato de serem dados comuns a todas as instituições de ensino. Os atributos selecionados foram as notas dos alunos no decorrer do período letivo.

Cavalcanti (2014) dividiu seu método em dois contextos, um utilizando uma abordagem genérica a partir de dados de um AVA, e a outra foi uma abordagem genérica a partir de um SCA (Sistema de Controle Acadêmico).

O AVA escolhido foi o *Moodle*, que é o AVA definido para o presente trabalho, os dados necessários para o processo de KDD foram obtidos através de consultas as tabelas mdl_user, mdl_log e mdl_grades, das quais foram extraídas as notas parciais dos estudantes agrupadas por atividades. Cavalcanti (2014) relata que através de algoritmos de classificação, aplicados nas notas das atividades iniciais, é possível prever se um aluno será aprovado ou reprovado na disciplina. Por sua vez, a partir da utilização dos dados classificados de todas as disciplinas no respectivo curso, é possível prever a evasão do aluno no curso de graduação. Para essa abordagem foi utilizado o algoritmo de Árvore de Decisão *48* já descrito na Seção 2.2.

Para a experimentação do segundo modelo, os dados utilizados foram obtidos do SCA

da UFPB Virtual (Unidade de Educação a Distância da Universidade Federal da Paraíba), que integra o sistema de Universidade Aberta do Brasil (UAB). Ele dividiu a sua base de dados em duas classes distintas, alunos graduados e alunos evadidos.

Para os testes realizados com o método de predição desenvolvido por Cavalcanti (2014), foram utilizados os seguintes algoritmos: *SimpleCart*, *J48* e o *ADTree*, que são algoritmos já descritos na Seção 2.2.

Para validação do método preditivo desenvolvido, eles utilizam o método de Acurácia Geral, que é utilizado para medir a proporção total dos estudantes com situação final, evadido ou graduado, que foi previsto pelas técnicas utilizadas. O critério é simples, é baseado na quantidade de alunos corretamente classificados na classe de graduados, com a quantidade de alunos corretamente classificados na classe de evadidos, divido pela quantidade total de alunos.

As técnicas utilizadas para a seleção dos atributos mais relevantes, como também os algoritmos de predição, e os métodos para avaliar a precisão dos resultados óbitos, influenciaram o presente trabalho. Porém, diferente de Cavalcanti (2014), o presente trabalho encapsulou todo o processo de mineração, predição e acompanhamento do aluno em Agentes desenvolvidos em *JADE*.

3.3 Minerando Dados Educacionais com foco na Evasão Escolar: oportunidades, desafios e necessidades

O trabalho desenvolvido por Rigo (2012), tem como principal objetivo justificar com base em seus estudos, a necessidade de uma ampliação no processo de análise inicial em relação aos fatores monitorados e que são utilizados na MDE, como também a inclusão de aspectos relacionados ao corpo docente e nas respectivas metodologias atribuídas a cada situação. De acordo com essa abordagem, a utilização e o desenvolvimento de soluções capazes de identificar precocemente perfis de alunos que possam evadir, é justificada, tendo como finalidade o apoio à pratica docente proporcionando um melhor acompanhamento dos discentes. Esse objetivo justifica e complementa o porquê que se faz necessária a utilização de soluções dinâmicas e inteligentes para o controle da evasão escolar apresentados no presente trabalho.

Rigo (2012) destaca em sua abordagem que os principais fatores para a evasão escolar são relacionados à aspectos pessoais e sociais existentes antes do ingresso no curso, como

também os relacionados com o contato acadêmico, as metodologias de aprendizagem utilizadas e a integração institucional.

Para a identificar variáveis associadas com o comportamento de evasão, faz-se necessária a utilização da MD, e como o presente trabalho, este também através da MD será possível a geração de modelos que promovam ações de diagnóstico precoce e encaminhamento de ações preventivas (RIGO; CAZELLA; CAMBRUZZI, 2012).

O sistema proposto por Rigo (2012) promove uma implementação que segue as seguintes etapas de processos: descoberta de conhecimento, registro de padrões de interesse, identificação de tendências conforme os padrões descobertos, aviso aos envolvidos, registros das ações realizadas e resultados obtidos. Essa abordagem foi utilizada em um estudo de caso que envolveu cursos de graduação, e utilizou um AVA como fonte de dados para a detecção de perfis com tendência a evasão. Para o estudo de caso, foram utilizados algoritmos de redes neurais. Para trabalhos futuros foi definida a análise de utilização de informações linguísticas em consonância com recursos de mineração, tendo como objetivo aproveitar melhor os dados não textuais disponíveis no AVA, assim permitindo aumentar as possibilidades de reconhecimento e comunicação de padrões significativos para o apoio a pratica docente.

O presente trabalho não utilizou algoritmos de redes neurais nos processos de KDD, porém tomou como influencia as justificativas apresentadas por Rigo (2012) para justificar o uso da MDE em prol do auxilio do acompanhamento educacional dos alunos, tendo como finalidade influencia-los a se dedicarem e recuperarem determinadas deficiências em seus estudos.

3.4 Minerando Dados sobre o desempenho de alunos de cursos de educação permanente em modalidade EAD: Um estudo de caso sobre evasão escolar na UNA-SUS

Da Costa (2012) demonstra em sua pesquisa que através da utilização de Extração de Conhecimento em Base de Dados foi possível identificar padrões que correspondem a evasão em cursos na modalidade EAD para profissionais da saúde. Os dados foram fornecidos pela UFCSPA (Universidade Federal de Ciências da Saúde de Porto Alegre) correspondentes a cursos de especialização na área da saúde.

Para a etapa de MD, Da Costa (2012) utilizou dados contidos em extensas planilhas, que continham o nome, a sede, o tutor, notas presenciais, notas EAD, notas de recuperação,

informações de acesso ao AVA e informações sobre o desempenho dos alunos. Após a etapa de seleção e pré-processamento dos dados, apenas as notas das avaliações finais e informações referente ao status do aluno foram utilizadas. A base de dados continha informações de 249 alunos da turma de 2013 da pós graduação *lato sensu*, que foram disponibilizados pela coordenação do curso.

Da Costa (2012) para a etapa de MD utilizou como ferramenta o *Weka*. O algoritmo utilizado foi o *J48* de árvore de decisão, que teve uma acurácia de 97,6% para o problema proposto. Para validação dos dados foi utilizado o técnica *cross-validation* utilizando o método *k-fold*, já descrita na Seção 2.2, que assumiu o valor de 10 *folds*.

Apesar de Da Costa (2012) não utilizar Sistemas Multiagentes para o acompanhamento do comportamento escolar dos alunos, às técnicas utilizadas para identificar as regras para classificação, o algoritmo utilizado e a forma como validou o seu modelo de dados, foram de grande influência para o presente trabalho. No caso do presente trabalho, utilizou o algoritmo *J48* apenas para que fosse possível visualizar as possíveis regras utilizadas para classificação, mas o algoritmo que foi utilizado para realizar a classificação dos novos dados históricos dos alunos foi o *Random Forest*.

3.5 Sistemas Multiagentes: mapeando a evasão na educação a distância

Wilges (2010) em seu trabalho, propõe um modelo conceitual preditivo de evasão na modalidade de EAD, que é construído seguindo uma arquitetura para Sistemas Multiagentes. Através dessa abordagem, espera-se que o problema da grande quantidade de alunos com risco a evasão seja bem identificado e visualizado, promovendo possíveis técnicas para ações preventivas.

Wilges (2010) justifica sua abordagem com a utilização de uma comunidade de agentes adaptável às estratégias definidas no contexto. O SMA proposto estará em constante aprendizagem para se adequar aos mais diversos contextos.

Para o desenvolvimento do SMA foi utilizado o *framework JADE* tendo como base as especificações FIPA para a comunicação entre os agentes, semelhante ao trabalho proposto nesta monografia.

Para facilitar identificação dos agentes necessários para o desenvolvimento do SMA,

Wilges (2010) utilizou a ferramenta *AgentTool*, já descrita na Seção 2.4. Nela foi utilizada a técnica de especificação de casos de uso, que também podem ser expressados por diagramas de sequência, para especificar a troca de mensagens entre os papéis no modelo de SMA descrito, ambas técnicas de modelagem já descritas na Seção 2.4. Primeiro foi definido o papel do sistema e logo em seguida foram definidos os passos necessários para que esse papel seja cumprido.

De acordo com Wilges (2010), os papeis definidos e os respectivos agentes foram:

1. Papéis:
    - Controlar Evasão
    - Observar Perfil do Estudante no AVA
    - Gerar Informações para a Instituição
2. Agentes:
    - Agente de Controle de Sessão
    - Agente de Desempenho
    - Agente de Participação
    - Agente de Frequência
    - Agente de Monitoramento
    - Agente de Informação

Cada agente foi baseado nas características gerais dos AVA's utilizados atualmente. O agente mais importante é o Agente de Monitoramento que é o responsável por identificar riscos de evasão de acordo com as informações passadas pelos demais agentes comunicar o Agente de Informação, que é responsável por avisar o corpo docente (WILGES et al., 2010). Para a presente pesquisa, essa arquitetura conceitual será muito importante para influenciar no desenvolvimento da arquitetura do modulo proposto, em como os agentes podem ser divididos e seus respectivos compromissos.

3.6 Sistema Tutor Inteligente baseado em Agentes na plataforma MOODLE para Apoio as Atividades Pedagógicas da Universidade Aberta do Piauí

O trabalho de Silva, Machado e Araújo (2014) teve como finalidade o desenvolvimento de um Sistema Tutor Inteligente para a plataforma *Moodle*, com o objetivo de auxiliar nas atividades pedagógicas da Universidade Aberta do Piauí (UAPI).

A implementação foi feita utilizando Agentes Inteligentes desenvolvidos na plataforma *JADE*. O agentes desenvolvidos foram:

1. Agentes de Perfil: Será responsável por captar o perfil do aluno, identificando suas deficiências e necessidades;

2. Agente de desempenho: Proporciona condições de decisão de que tarefa ou ação a ser executada;

3. Agente Comunicador: Servirá de elo entre processo do STI e o tutor, colocando este a par das atividades exercidas pelos alunos e sugerindo intervenções pedagógicas.

Para a descoberta de padrões nos dados obtidos através das interações dos usuários no *Moodle* foi utilizado o algoritmo *k-means*, que é um algoritmo de agrupamento/clusterização, através dele foi possível dividir os alunos em grupos (*clusters*), tornando possível a recomendação de atividades pedagógicas para cada perfil. O algoritmo *k-means* utiliza um parâmetro de entrada k, que determina a quantidade de *clusters* (coleção de objetos que são similares uns aos outros (de acordo com algum critério de similaridade pré definido) e dissimilares a objetos pertencentes a outros *clusters*), sendo que tais *clusters* possuem n elementos (os *clusters* podem ter quantidade de elementos diferentes). Para que fosse possível classificar as novas instancias dos dados dos usuários em seus respectivos grupos, foi utilizado o algoritmo *J48*, já descrito na Seção 2.2. Ambos os algoritmos foram utilizados através da ferramenta para mineração de dados *Weka*.

O presente trabalho assemelha-se bastante com o de Silva, Machado e Araújo (2014), a maior diferença é no algoritmo escolhido para a classificação das novas instancias dos dados dos alunos, que é o *Random Forest*, ele demonstrou ser mais eficiente tendo uma maior acurácia de acordo com o modelo de dados desenvolvido no presente trabalho, que será descrito no capitulo 4. Outra diferença é na arquitetura multiagente definida para o presente trabalho, que

também será descrita no capitulo 4. Apesar das diferenças o presente trabalho se influencia da abordagem de Silva, Machado e Araújo (2014) para a descoberta das classes que servirão para classificar os alunos. O presente trabalho também utilizou o algoritmo *k-means* e separou os dados em 5 *clusters* (MUITO RUIM, RUIM, REGULAR, BOM e MUITO BOM).

A Tabela 1 apresenta um comparativo entre as características dos trabalhos relacionados e o presente trabalho.

Tabela 1 – Comparativo das Características dos Principais Trabalhos Relacionados

Trabalho	Moodle	Multiagente	Aprendizagem	Técnica
(KAMPFF, 2009)			Supervisionada	Classificação
(CAVALCANTI et al., 2014)		X	Supervisionada	Classificação
(RIGO; CAZELLA; CAMBRUZZI, 2012)			Não Supervisionada	Redes Neurais
(COSTA; CAZELLA; RIGO, 2012)	X		Supervisionada	Classificação
(WILGES et al., 2010)		X	Não se Aplica	Não se Aplica
(SILVA; MACHADO; ARAÚJO, 2014)	X	X	Não Supervisionada	Clusterização
Este Trabalho	X	X	Não Supervisionada	Clusterização

4 PROCEDIMENTOS METODOLÓGICOS E RESULTADOS

Este capítulo descreve os procedimentos metodológicos que foram necessários para a conclusão deste trabalho.

4.1 Análise do SMA desenvolvido pelo Grupo de Estudo de Engenharia de Software em Sistemas Multiagente (GESMA)

O *SMA Moodle* foi desenvolvido utilizando o *framework JADE*, uma extensão denominada *JAMDER* e respeitando os padrões do protocolo de comunicação *FIPA*, que são tecnologias já descritas na Seção 2.4.

O SMA interage com os dados do *Moodle* através do acesso ao banco de dados deste AVA. Através da captura das informações do banco de dados, o SMA atualiza as informações acessíveis aos agentes. Os agentes podem postar mensagens em fóruns, utilizam *chats*, criação de links ou arquivos no ambiente.

Os agentes já em funcionamento no sistema são: agente companheiro de aprendizagem, agente pedagógico, agente acompanhante de tutores, agente fornecedor de materiais, agente formador de grupos e o agente auxiliar de usabilidade. Os agentes citados anteriormente serão descritos com mais detalhes logo a seguir, explorando suas principais características e responsabilidades de acordo com Gonçalves et al. (2014):

1. **Agente Companheiro de Aprendizagem**: O Agente Companheiro de Aprendizagem é responsável por auxiliar o processo de aprendizagem dos alunos no decorrer do curso. De acordo com o desempenho do aluno, o agente envia mensagens para eles. Estas mensagens podem conter informações de apoio, reforço ou sugestões de atividades.

2. **Agente Pedagógico**: O Agente Pedagógico tem a responsabilidade de acompanhar e dar sugestões aos usuários em seus respectivos cursos, disciplinas e projetos.

3. **Agente Acompanhante de Tutores**: O Agente Acompanhante de Tutores é o agente responsável por acompanhar e monitorar os tutores em seus respectivos cursos dando sugestões sobre materiais de estudo antes das disciplinas começarem, como também dicas em relação a participação do mesmo em fóruns e na postagem de material didático.

4. **Agente Fornecedor de Materiais**: Este agente tem como principal característica enviar conteúdo digital de acordo com a disciplina especificada. Seu trabalho é realizado em conjunto com o Agente Companheiro de Aprendizagem. De acordo com o desempenho do aluno, caso o seu rendimento torne-se baixo ao decorrer do curso, este agente se responsabiliza de enviar material complementar por meio de postagens de arquivos ou *link* no *Moodle*, como também por meio de *e-mails* e *twitters*.

5. **Agente Formador de Grupos**: Este agente é responsável por criar grupos de usuários de acordo com características que indiquem afinidade, como perfis, temas e aprendizagem. Como o Agente Fornecedor de Materiais, ele também trabalha junto ao Agente Companheiro de Aprendizagem.

6. **Agente Auxiliar de Usabilidade**: O Agente Auxiliar de Usabilidade é responsável por auxiliar os novos usuários como também os veteranos em relação a possíveis dificuldades dos mesmos em relação ao AVA.

Os agentes descritos correspondem ao corpo principal do SMA em questão. Sua execução acontece em conjunto ao *Moodle* e os seus dados são alimentados de acordo com os dados contidos na base de dados do AVA.

4.2 Análise da Base de Dados do Moodle da Universidade Estadual do Ceará (UECE) e Seleção de Algoritmos para Mineração

O AVA *Moodle* é um sistema modular, que possui o gerenciamento de vários módulos voltados ao gerenciamento dos cursos. A sua estrutura relacional do banco de dados reflete essa característica.

4.2.1 Análise

As tabelas no banco de dados são compostas pelo prefixo e nome do módulo. O prefixo padrão é o *mdl_*. Isso pode ser alterado no momento da instalação. Por exemplo, a tabela do módulo Curso é mdl_course. Todos os módulos seguem esse padrão.

Cada módulo possui uma tabela principal e tabelas secundárias. A estrutura da nomenclatura dessas tabelas são:

1. Tabela Principal: *mdl_* + nome do modulo. Ex: mdl_course.
2. Tabela Secundaria: *mdl_* + nome do modulo + funcionalidade do módulo. Ex: mdl_course_categories.

Vale ressaltar que alguns conceitos das nomenclaturas dos componentes do *Moodle* diferem dos utilizados comumente nas Instituições de Ensino. Um exemplo comum é sobre o módulo de cursos. Na verdade cursos para o *Moodle* são disciplinas para uma Universidade, por sua vez, esse conjunto de cursos fazem parte de um grupo, que correspondem a matriz curricular de disciplinas que compõem um curso para a Instituição de Ensino.

Para o presente trabalho, os módulos mais importantes foram:

1. ***User***: para a captura dos alunos matriculados na plataforma.
2. ***Course***: para obter os cursos registrados no banco de dados e em quais cursos cada aluno esta matriculado.
3. ***Log***: para capturar os dados de participação do aluno
4. ***Grades***: para obter as notas das avaliações dos alunos.

4.2.2 Seleção

Os dados fornecidos para o presente trabalho correspondem a uma quantidade de 3195 alunos, 248 cursos e de um intervalo de tempo de Agosto de 2011 à Agosto de 2013.

No presente trabalho de acordo a analise do banco de dados e estudos dos trabalhos relacionados, foi possível selecionar as informações mais importantes que correspondessem às interações dos usuários na plataforma para a construção do modelo de dados em questão. Elas foram escolhidas por serem os valores quantitativos que mais são utilizados pelos alunos. As informações elencadas no processo de Seleção dos Dados foram:

1. Identificação do Aluno
2. Identificação do Curso

3. Data de Criação do Curso
4. Data que o Curso Iniciará
5. Média Final de Cada Curso
6. Período Semanal
7. Período Mensal
8. Período Semestral
9. Quantidade de Acessos ao Curso
10. Quantidade de Acessos ao Fórum
11. Quantidade de Postagens no Fórum
12. Quantidade de Atividades Entregues
13. Média das Notas das Atividades
14. Quantidade de Acessos aos Arquivos
15. Quantidade de Acessos às Wikis

4.2.3 Pré-Processamento

Para a construção do modelo, foram utilizados atributos que passaram por pré-processamento para que fossem obtidos, visto que as informações contidas neles não estavam de forma clara (como o modelo precisava) e organizadas no banco de dados do *Moodle*. Para isso foi criado um *Data Mart* que será alimentado mensalmente, e que também servirá para guardar o histórico de acompanhamento dos alunos ao decorrer do período acadêmico.

Para a etapa de pré-processamento, foi desenvolvida uma aplicação na linguagem *Java*[1] utilizando *JDBC*[2] no ambiente de desenvolvimento *Eclipse*[3]. Essa aplicação mapeia o banco de dados do *Moodle* tendo como finalidade capturar as informações necessárias para alimentar o *Data Mart*.

[1]Disponível em: https://www.java.com/pt_BR
[2]Disponível em: http://www.oracle.com/technetwork/java/javase/jdbc/index.html
[3]Disponível em: https://eclipse.org

4.2.4 Organização e Alimentação do Data Mart

O *Data Mart* é composto por um conjunto de tabelas que suprem às informações necessárias para a construção e atualização do modelo de dados do presente trabalho, como também para o gerenciamento do acompanhamento semestral dos alunos que é realizado pelo modulo de evasão desenvolvido nesse trabalho. Inicialmente ele foi alimentado com os dados históricos dos alunos contidos no intervalo de Agosto de 2011 a Agosto de 2013, para que um modelo de dados inicial fosse definido. Os dados foram obtidos de todos os cursos registrados no banco de dados que tiveram registros dos atributos destacadas na etapa de seleção. A seguir será descrito sucintamente cada uma das tabelas e suas finalidades.

1. ***Semester***: tabela que é registrado cada semestre letivo, para que tenha o controle da captura das informações correspondentes a cada semestre.

2. ***Course***: a cada semestre é registrado os cursos ofertados para os alunos.

3. ***Week Control***: é dividido para cada curso um agrupamento de dados em janelas de tempo para que os dados de participação de cada aluno seja capturado. Aproximadamente são divididos em 19 semanas que correspondem a um semestre letivo.

4. ***Month Control***: para cada curso é registrado um período mensal para que aconteça a classificação do status daquele aluno. Por *default* é configurado para registrar bimestralmente ou conforme seja configurado pelo administrador do sistema.

5. ***Student e Student Course***: a tabela *Student* é utilizada para salvar todos os alunos matriculados do *Moodle* e referencia-los com seus respectivos cursos com a tabela *Student Course*.

6. **Historic**: é a tabela mais importante, onde serão registrados os dados semanalmente de cada aluno em relação aos cursos matriculados naquele semestre. O atributo *register_type* corresponde ao tipo de registro que está sendo armazenado. Os tipos de registro são:

    - QAC: Quantidade de Acessos ao Curso
    - QAF: Quantidade de Acessos ao Fórum
    - QPF: Quantidade de Postagens no Fórum

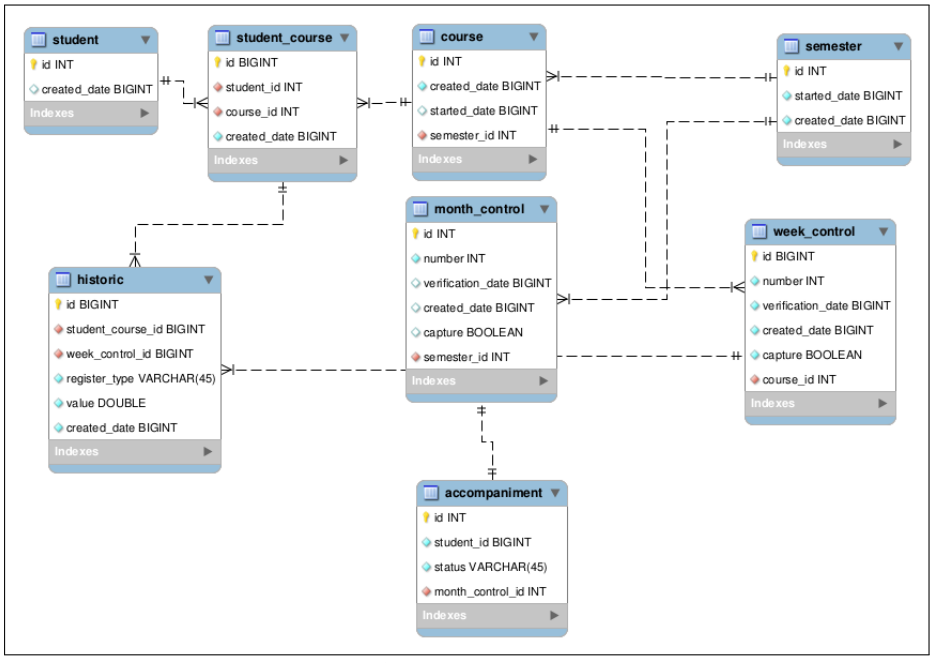
- QAA: Quantidade de Acessos aos Arquivos
- QAW: Quantidade de Acessos às Wikis
- QAE: Quantidade de Atividades Entregues
- MNA: Média das Notas das Atividades

7. ***Accompaniment***: esta tabela será alimentada com a verificação bimestral dos alunos pelo processo de classificação, que será explicado na próxima subseção. O atributo *status* corresponde aos grupos em que os alunos são divididos em relação aos seus dados quantitativos, os valores que esse atributo pode assumir são:

   - MUITO BAIXO RISCO
   - BAIXO RISCO
   - REGULAR
   - RISCO
   - FORTE RISCO

A figura 3 ilustra o modelo Entidade Relacionamento da entidades que contemplam o *Data Mart*.

Figura 3 – Modelo Entidade Relacionamento do *Data Mart*



Fonte: fornecido pelo autor

4.3 Análise comparativa para a escolha de uma Ferramenta que auxilie o processo de Extração de Conhecimento

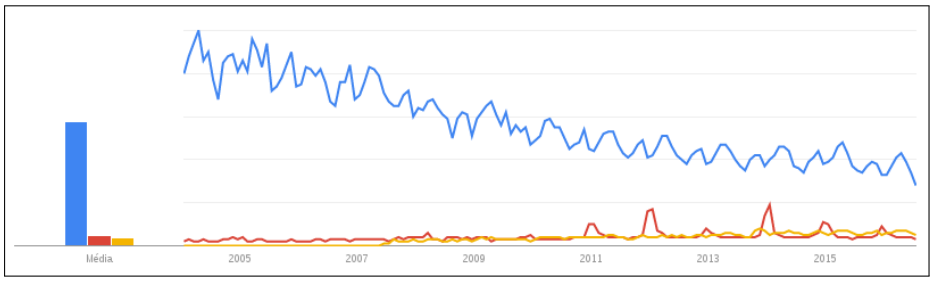
Foi realizado uma analise comparativa entre três ferramentes que podem auxiliar no processo de KDD. As ferramentas são:

1. RapidMiner
2. Weka
3. Elki

Para a escolha da ferramenta ideal foram definidas alguma métricas, das quais a que foi mais relevante foi, se a ferramente possui *API* para ser utilizados seus métodos de descoberta de conhecimento em *Java*, pelo fato do modulo desenvolvido no presente trabalho ser nessa linguagem, visto que os agentes que serão explicados na Seção seguinte, encapsulam todo o processo de KDD.

Além de analisar se as ferramentas possuíam *API*, foram verificados a popularidade e documentação das mesmas. O gráfico 4 da popularidade das ferramentas aqui citadas.

Figura 4 – Comparativo da Popularidade das Ferramentas



Fonte: Gerado no Google Trends

No gráfico 4 a cor azul representa a popularidade da ferramenta *Weka*, a cor vermelha representa a popularidade da ferramenta *Elki*, e por sua vez, a cor amarela representa a popularidade da ferramenta *RapidMiner*, diferente das outras ferramentas já citadas, o *RapidMiner* não é *open source*. Os dados foram comparados com informações obtidas de 2005 a 2015, em relação as buscas relacionadas às ferramentas no buscador do *Google*[4].

A ferramenta escolhida foi a *Weka*, por conter uma boa documentação e interface agradável de manipulação, proporcionando assim uma curva mínima de aprendizagem.

[4]Disponível em: http://google.com

4.3.1 Modelo, Mineração dos Dados e Descoberta de Padrões

Para a construção do Modelo foi utilizada a *API* fornecida pela ferramenta *Weka*. Através dela foi possível utilizar na linguagem *Java* os métodos necessários os passos a seguir. Após o *Data Mart* ser alimentado com os dados históricos dos alunos que já estão cursaram, os dados são agrupados por aluno em um arquivo *arff*, essa escolha é derivada da eficiência que é obtida utilizando ele em conjunto com a *API* do *Weka*, visto que tipo de arquivo foi desenvolvido especificamente para ser usado pela ferramenta. Cada instancia do modelo de dados é composta pelos seguintes atributos:

Tabela 2 – Descrição dos Dados que compõem o Modelo Inicial

Atributo	Descrição	Tipo
QAC	Quantidade de Acessos ao Curso	Numérico
QAF	Quantidade de Acessos ao Fórum	Numérico
QPF	Quantidade de Postagens no Fórum	Numérico
QAE	Quantidade de Atividades Entregues	Numérico
MNA	Média das Notas das Atividades	Numérico
QAA	Quantidade de Acessos aos Arquivos	Numérico
QAW	Quantidade de Acessos às Wikis	Numérico

Para que fosse possível dividir os alunos em grupos de acordo com seus dados quantitativos capturados ao decorrer do tempo, foi utilizado um algoritmo aprendizagem não supervisionada, o *K-Means*, a escolha desse algoritmo para esse contexto foi influenciada pelo trabalho de Silva, Machado e Araújo (2014), que se assemelha com o problema abordado nesse trabalho. O *K-Means* é um algoritmo de clusterização, já explicado na Seção 3.6. Após o processo de clusterização no modelo de treinamento, os dados foram divididos em 5 grupos (MUITO BAIXO RISCO, BAIXO RISCO, REGULAR, RISCO e FORTE RISCO). O fator determinante para que os grupos fossem divididos da seguinte forma, foi o atributo de média das notas das avaliações realizadas (MNA). Cada grupo ficou dividido da seguinte forma:

1. MUITO BAIXO RISCO: 9% dos dados equivalente a 298 alunos.
2. BAIXO RISCO: 19% dos dados equivalente a 618 alunos.
3. REGULAR: 20% dos dados equivalente a 641 alunos.
4. RISCO: 15% dos dados equivalente a 472 alunos.

5. FORTE RISCO: 36% dos dados equivalente a 1166 alunos.

De acordo com os atributos utilizados para a construção do modelo clusterizado, a tabela a seguir demonstra os valores iniciais de cada atributo que determina a entrada da instancia no *cluster* correspondente, que é encontrado após executar o algoritmo *K-Means* no *dataset* atual. Foi utilizada a técnica distância euclidiana como método do calculo de similaridade, que é uma técnica que mede a distancia entre dois pontos, essa escolha não teve nenhum motivo especifico, pois não foi testado outras técnicas para calcular a similaridade e comparar os resultados, a técnica Distância Euclidiana é configurada por *default* no *Weka*. Vale ressaltar, que esses valores podem mudar ao longo do tempo, conforme novos dados vão sendo inseridos no *data mart*. A figura 5 ilsutra uma pequena parte do arquivo *arff* gerado e o gráfico 7 ilustra a variância de cada atributo em relação aos seus valores.

Figura 5 – Visualização de uma pequena parte do Arquivo *Arff* gerado após a etapa de clusterização

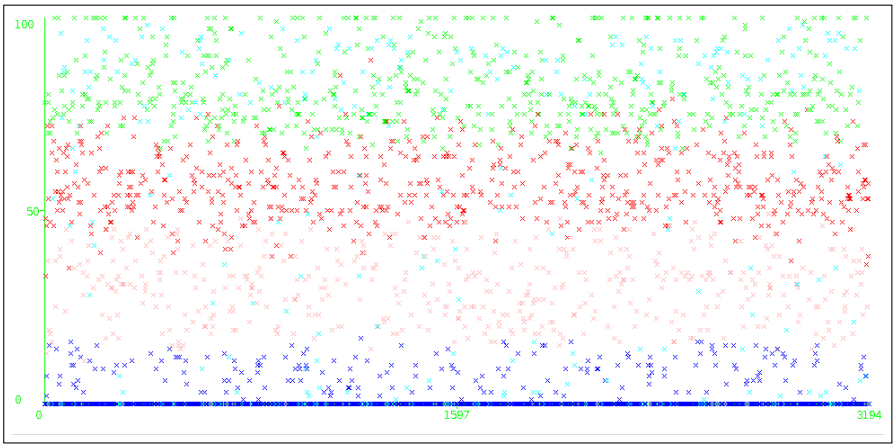
No.	1: Instance_number	2: qae	3: qac	4: qaf	5: mna	6: qpf	7: qaa	8: qaw	9: **Cluster**
	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Nominal
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	cluster0
2	1.0	15.0	64.0	40.0	78.0	7.0	2.0	11.0	cluster2
3	2.0	0.0	10.0	4.0	0.0	0.0	12.0	0.0	cluster0
4	3.0	0.0	9.0	0.0	0.0	0.0	0.0	0.0	cluster0
5	4.0	3.0	15.0	2.0	48.0	0.0	14.0	0.0	cluster1
6	5.0	15.0	99.0	9.0	33.0	0.0	5.0	0.0	cluster1
7	6.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	cluster0
8	7.0	2.0	2.0	7.0	2.0	3.0	0.0	0.0	cluster0
9	8.0	4.0	45.0	12.0	46.0	2.0	5.0	0.0	cluster1
10	9.0	2.0	2.0	1.0	7.0	0.0	2.0	0.0	cluster0
11	10.0	3.0	27.0	4.0	13.0	0.0	23.0	0.0	cluster4
12	11.0	1.0	9.0	1.0	37.0	0.0	10.0	0.0	cluster4
13	12.0	3.0	27.0	3.0	72.0	0.0	42.0	0.0	cluster1
14	13.0	1.0	3.0	1.0	0.0	1.0	0.0	0.0	cluster0
15	14.0	4.0	49.0	23.0	53.0	4.0	6.0	0.0	cluster1
16	15.0	4.0	49.0	26.0	80.0	2.0	9.0	0.0	cluster2
17	16.0	12.0	91.0	81.0	78.0	5.0	1.0	52.0	cluster2
18	17.0	0.0	0.0	0.0	70.0	0.0	0.0	0.0	cluster2
19	18.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	cluster0
20	19.0	0.0	9.0	2.0	15.0	0.0	0.0	0.0	cluster0
21	20.0	3.0	26.0	18.0	19.0	1.0	7.0	0.0	cluster4
22	21.0	0.0	0.0	0.0	70.0	0.0	0.0	0.0	cluster2
23	22.0	2.0	32.0	20.0	47.0	1.0	6.0	0.0	cluster1
24	23.0	1.0	11.0	2.0	18.0	0.0	5.0	0.0	cluster4
25	24.0	2.0	12.0	8.0	73.0	1.0	2.0	0.0	cluster2
26	25.0	0.0	3.0	1.0	0.0	0.0	2.0	0.0	cluster0
27	26.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	cluster0
28	27.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	cluster0
29	28.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	cluster0
30	29.0	2.0	41.0	12.0	16.0	1.0	3.0	0.0	cluster4
31	30.0	1.0	25.0	3.0	65.0	0.0	11.0	0.0	cluster1
32	31.0	7.0	62.0	3.0	72.0	0.0	38.0	0.0	cluster1
33	32.0	26.0	108.0	126.0	53.0	18.0	4.0	81.0	cluster3

Autor: Fornecido pelo Autor

Tabela 3 – Valores iniciais que determinam em qual *cluster* cada instancia corresponde

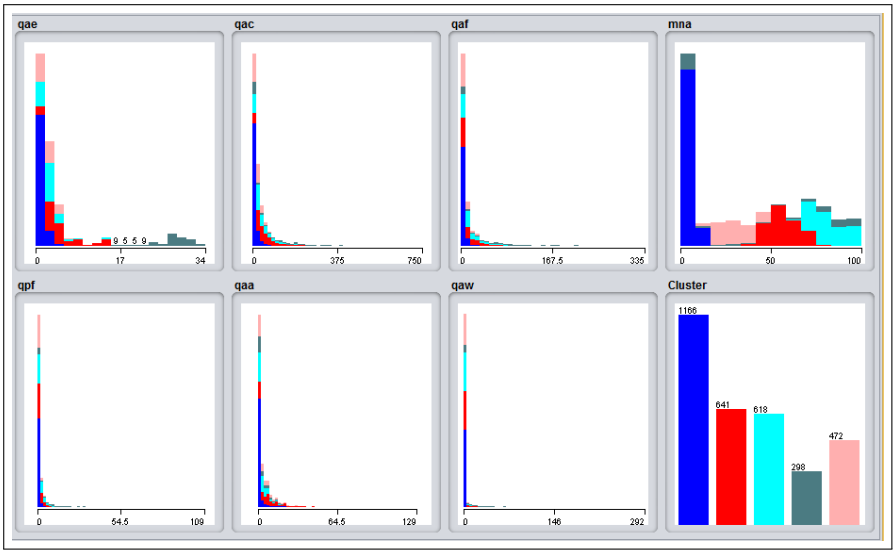
Cluster	QAE	QAC	QAF	MNA	QPF	QAA	QAW
MUITO BAIXO RISCO	30	307	299	92	101	28	86
BAIXO RISCO	9	148	82	86	9	6	21
REGULAR	3	65	14	77	0	21	0
RISCO	1	8	3	19	0	3	0
FORTE RISCO	0	0	0	0	0	0	0

Figura 6 – Mapa de calor dos *clusters*, onde cada cor representa um *cluster* em ordem crescente



Autor: Fornecido pelo Autor

Figura 7 – Gráfico de variância dos valores dos atributos



Autor: Fornecido pelo Autor

Na figura 6 o eixo X representa a quantidade de alunos, o eixo Y representa a faixa de valores do atributo *MNA*. A cor azul escura representa o *cluster* FORTE RISCO, a rosa o

RISCO, o vermelho o REGULAR, o verde o BAIXO RISCO e o azul claro o MUITO BAIXO RISCO.

Os alunos que foram alocados aos grupos RISCO e FORTE RISCO, são alunos que possuem mal desempenho em suas interações com a plataforma, sendo fortes concorrentes a desistirem do curso. Os alunos que são classificados com esses *status* são os que sofrem intervenção do SMA para melhorarem seu desempenho educacional.

Com a descoberta das classes foi realizado um teste comparativo entre cinco algoritmos e avaliado o que teve a maior acurácia em relação aos dados do modelo. Os algoritmos testados foram: *SimpleCart*, *J48*, *JRip* (equivalente ao RuleLearner) e *Random Forest*.

A tabela a seguir apresenta a acurácia de cada algoritmo em relação ao modelo de dados. Para a verificação foi utilizada a técnica *cross-validation*, já explicada na Seção 2.2. Os dados foram divididos em 10 *folds*.

Tabela 4 – Comparação entre os algoritmos de Classificação em relação a Acurácia

Algoritmo	Acurácia
SimpleCart	97,24%
J48	97,32%
JRip	96,95%
Random Forest	98,27%

Através da comparação de acurácia dos algoritmos em relação ao modelo de treinamento, o *Random Forest* foi o que obteve o melhor resultado, sendo assim o escolhido para ser utilizado no módulo desenvolvido no presente trabalho. A acurácia dos algoritmos foi determinada de acordo com a Métrica de *Kappa*, que é uma forma de medir a concordância das interpretações dos valores de cada instancia do modelo.

4.3.2 Atualização do Modelo

Ao final de cada semestre letivo, os dados históricos capturados dos alunos serão integrados com os dados históricos utilizados para a definição do modelo de dados inicial. Após isso, o modelo passará novamente pelo processo de clusterização, para que as classes sejam atualizadas e o modelo para predição fique cada vez mais inteligente acompanhando os padrões de interação dos alunos com a plataforma.

4.4 Desenvolvimento da Arquitetura do Módulo de Evasão do SMA do GESMA

Para o presente trabalho foi possível identificar alguns papéis que o módulo em questão deve suprir. Esses papéis são:

1. Controlar o Tempo em que um Semestre Inicia e Termina, para capturar os dados e alimentar o *Data Mart* periodicamente.

2. Capturar os Dados periodicamente das interações dos alunos no *Moodle*.

3. Classificar os alunos bimestralmente.

4. Comunicar os demais agentes do ambiente sobre o que foi interpretado com a classificação dos alunos, para que os demais agentes tomem decisões para auxiliar o aluno a melhorar o desempenho e terem êxito nos cursos matriculados.

Para que fosse possível atender esses papéis foi desenvolvido o Agente Controlador de Evasão. Este Agente possui encapsulado em seus comportamentos todo o processo de KDD descrito na Seção 4.2, como também a comunicação com os demais agentes do SMA, principalmente com o Agente Companheiro de Aprendizagem, cujo suas características já foram descritas na Seção 4.1.

O fluxo dos processos executados pelos comportamentos do Agente Controlador de Evasão será descrito a seguir:

1. Verificação se a data atual de execução está dentro do intervalo de tempo correspondente ao semestre atualmente cadastrado.

2. Caso a data não esteja entre o intervalo de tempo correspondente ao semestre atualmente cadastrado, o comportamento do Agente Controlador de Evasão se encerra nesse ciclo.

3. Caso a data atual esteja dentro do intervalo de tempo, ele irá verificar os cursos que foram registrados para serem ofertados durante aquele semestre letivo.

4. Com a identificação dos Cursos, ele irá registrar no *Data Mart* o período semanal que deverá ocorrer a captura dos dados.

5. Analogamente ao período semanal, ele irá registrar o período mensal em que deverá acontecer a classificação dos alunos para obter a cada etapa o estado do desempenho do aluno.

6. Se a data atual corresponder a algum dia que deverá ocorrer a captura dos dados, o Agente irá capturar os dados de cada aluno por curso, em um intervalo de uma semana. Os dados capturados semanalmente já foram descrito conforme na Seção 4.3.1.

7. Se a data atual corresponder a algum dia que deverá ocorrer a classificação dos dados dos alunos, o Agente irá agrupar os dados capturados do inicio do semestre até aquela data, e irá classificar o aluno utilizando o algoritmo *Random Forest* que estará encapsulado dentro do seu comportamento através da *API* do *Weka*. O modelo que servirá como base para classificar o *status* desse aluno, é o modelo construído após o processo de clusterização.

8. Identificando os alunos que possuem o *status* referente aos grupos de alunos RISCO e FORTE RISCO, serão encaminhados para o Agente Companheiro de Aprendizagem, para que ele se responsabilize em auxiliar esses alunos a mudarem esse cenário de mau desempenho que tendem a evasão do curso e enviará por e-mail a relação de alunos com essa classificação para os professores/tutores e coordenadores.

9. Por fim, ao chegar no final do semestre corrente, o Agente irá agrupar todos os dados históricos de todos os alunos até o momento, executar o processo de clusterização para dividir os alunos nos grupos definidos na Seção 4.3.1, assim atualizando o modelo de predição de dados.

Após a definição dos papéis e de como se comunicar com os Agentes necessários, foi iniciada a etapa de implementação do Agente Controlador de Evasão.

4.5 Implementação do Sistema

O Agente Controlador de Evasão foi desenvolvido utilizando o *framework JADE* com a extensão *JAMDER*, de acordo com as definições descritas na subseção anterior. No Agente foi definido um comportamento cíclico, que é executado uma vez por dia. Nele estão todos os passos de execução descritos na subseção anterior.

A interação do Agente Controlador de Evasão com o banco de dados do *Moodle* e com o *Data Mart* foi desenvolvida utilizando *JDBC* com o banco de dados *PostgreSQL*[5]. Para as

[5]Disponível em: https://www.postgresql.org/

interações do Agente Controlador de Evasão com o banco de dados do *Moodle*, foram aproveitados os métodos já implementados no SMA. A interface de interação entre o SMA e o banco de dados do *Moodle* foi desenvolvida utilizando o *framework Hibernate* [6], que é um *framework* que realiza o mapeamento objeto relacional *Object Relational Mapper* (ORM), tendo como finalidade diminuir a complexidade do desenvolvimento da persistência dos dados.

4.6 Execução, Coleta de Dados e Testes

Para esta etapa o Agente Controlador de Evasão foi executado em um ambiente separado do SMA simulando um ambiente real de atividade, a fim de observar seu comportamento em relação a gerencia do processo de KDD nele encapsulado. Como já descrito na Seção 4.3.1, foram utilizados para testes os dados de 3195 alunos, 248 cursos e de um intervalo de tempo de Agosto de 2011 à Agosto de 2013. Após o Agente alimentar o *Data Mart* com as informações necessárias para construir o modelo de treinamento, o modelo foi submetido a um processo de clusterização, como já descrito na Seção 4.3.1, utilizando o algoritmo *K-means*, a fim de descobrir as classes necessárias para a predição. Após a descoberta dos grupos, o modelo foi submetido ao processo de cross-validation com o algoritmo *Random Forest*, validando a predição obtendo uma acurácia de 98,27%.

4.7 Análise e Validação dos Resultados Obtidos

Com o *Data Mart* alimentado pelo Agente, a predição foi testada e validada de duas formas:

1. Através da Técnica de *Cross-validation*, já descrita na Seção 2.2, no qual com o algoritmo *Random Forest* foi obtido uma acurácia de 98,27%.

2. Foi fornecido pela UECE uma planilha contendo a relação de alunos divididos entre alunos graduados, desistentes, que abandonaram e que foram transferidos do curso.

A planilha continha a relação de alunos já matriculados no período de 2006 a 2014 nos

[6] Disponível em: http://hibernate.org/

cursos ofertados pela UAB/UECE, totalizando 3359 alunos, dos quais 405 são alunos desistentes. Apesar da planilha ser de um espaço de tempo maior do que o dos dados encontrados no *Moodle*, foi possível identificar em 92.5% dos alunos que poderiam evadir resultantes da predição em comparação com os alunos desistentes apontados na planilha.

Através dessas informações reais dos alunos, houve uma comparação com o histórico de predição. Houve casos em que alunos considerados bons pela etapa de classificação foram identificados como alunos desistentes ou que abandonaram o curso, esse tipo de caso é impossível prever, pois provavelmente partiu de uma causa externa em relação ao aluno e sua vida pessoal. Em contra partida todos os alunos que tiveram como causas de suas desistências apenas problemas em relação com a plataforma, foi possível prever com a técnica de classificação.

5 CONCLUSÃO E TRABALHOS FUTUROS

A Educação a Distância vem crescendo com o apoio de ferramentas digitais. Através dos AVA's foi possível aumentar a escala de alcance de usuários, porém acompanhar esses alunos é uma atividade com tendencia a falhas, mas que pode ser auxiliado com o uso de ferramentas computacionais autônomas como Sistemas Multiagentes(SMA's). Através da Mineração de Dados Educacionais, tornou-se possível a descoberta de padrões de comportamento que refletem o desempenho do aluno. Agrupar e interpretar essas informações é de extrema importância para as instituições de ensino.

Neste trabalho foi apresentado o desenvolvimento de um módulo utilizando tecnologia Multiagente e Mineração de Dados, para ser integrado ao SMA desenvolvido pelo grupo GESMA da Universidade Federal do Ceará. Esse módulo tem como finalidade a descoberta de padrões para que seja possível identificar alunos com tendencia a mau desempenho e evasão escolar. Por sua vez, interagindo com os demais agentes do SMA para alterar o cenário dos alunos encontrados com baixo desempenho.

O módulo foi desenvolvido utilizando o *framework JADE + JAMDER*, que teve como resultado um agente denominado Agente Controle de Evasão, que possui encapsulado em seus comportamentos processos de descoberta de conhecimento, se beneficiando de aprendizagem supervisionada para a predição dos dados com o algoritmo *Random Forest* e aprendizagem não supervisionada para identificar às classes através de clusterização com o algoritmo *K-Means*, dividindo os alunos em cinco grupos de acordo com seus índices de participação na plataforma *Moodle*. Os cinco grupos citados anteriormente são: alunos com BOA participação, MUITO BOA participação, participação REGULAR, participação RISCO e FORTE RISCO. Identificando os alunos correspondentes a esses grupos, foi possível realizar a elaboração de um modelo de dados para predição, o qual foi utilizado para acompanhar o desempenho dos alunos ao decorrer do semestre letivo, tendo como finalidade remediar os possíveis alunos com tendencia a evasão, a fim de os ajudarem a melhorar o desempenho, através dos comportamentos do Agente Companheiro de Aprendizagem implementado no SMA.

Através da clusterização, foi possível observar a imensa quantidade de alunos que estão com desempenho ruim ou muito ruim nos cursos. Os dados desses alunos totalizam 51% de todo o *dataset*, o que é algo preocupante, sendo possível concluir que é necessário melhorar o acompanhamento dos alunos, dessa forma utilizando sistemas computacionais autônomos como

Sistemas Multiagentes e Mineração de Dados Educacionais.

O presente trabalho resultou em um módulo para predição de alunos com tendência a evasão na plataforma *Moodle*, além dele ser utilizado como uma parte integrante do SMA desenvolvido pelo grupo GESMA, ele também pode ser utilizado individualmente, sendo necessário o desenvolvimento de uma interface gráfica para a interpretação visual dos dados. Porém, o *Data Mart* está organizado de uma forma que está pronta para fornecer diversos relatórios.

Além da definição e desenvolvimento do processo de KDD para o presente contexto, o trabalho contribuiu com uma analise de algoritmos de aprendizagem supervisionada, comparando-os em relação a acurácia, o resultado foi obtido através da técnica *cross-validation* no modelo de treinamento gerado após o processo de clusterização para a descoberta das classes, dos quais foi possível observar dentre os selecionados, o mais eficiente foi o *Random Forest*.

O SMA não foi testado em ambiente real com o módulo de Controle de Evasão, assim não sendo possível observar mudanças reais em relação ao desempenho dos alunos, porém com o processo de KDD validado, tendo uma acurácia significativa de 98,27% de acerto, foi possível identificar precocemente alunos que tendem a evasão, dessa forma acionando os Agentes necessários para mudar esse cenário.

Como trabalhos futuros podemos citar uma analise para fragmentar em mais Agentes as funcionalidades contidas nos comportamentos do Agente Controle de Evasão, o desenvolvimento de uma interface gráfica para configuração dos parâmetros necessários para o funcionamento do módulo, como também a visualização mais detalhada das informações que podem ser obtidas através do *Data Mart* que foi definido no presente trabalho. E por fim, o estudo da viabilidade da migração do *Data Mart* do banco de dados relacional para um banco de dados não relacional, visando o ganho de desempenho para a manipulação de imensas cargas de dados.

REFERÊNCIAS

BATISTA, A. F. d. M. Desenvolvendo sistemas multiagentes na plataforma jade. *Santo André: Universidade Federal do Abc*, p. 32, 2008.

BREIMAN, L. Random forests. *Machine learning*, Springer, v. 45, n. 1, p. 5–32, 2001.

BREIMAN, L. et al. *Classification and regression trees*. [S.l.]: CRC press, 1984.

CAMBRUZZI, W. L. Gvwise: uma aplicação de learning analytics para a redução da evasão na educação a distância. Universidade do Vale do Rio dos Sinos, 2014.

CAVALCANTI, Â. G. G. et al. Mineração e visualização de dados educacionais: Identificação de fatores que afetam a motivação de alunos na educação a distância. 2014.

COHEN, W. W. Fast effective rule induction. In: *Proceedings of the twelfth international conference on machine learning*. [S.l.: s.n.], 1995. p. 115–123.

COSTA, S. S. da; CAZELLA, S.; RIGO, S. J. Minerando dados sobre o desempenho de alunos de cursos de educação permanente em modalidade ead: Um estudo de caso sobre evasão escolar na una-sus. *RENOTE*, v. 12, n. 2, 2012.

FAYYAD, U. M. et al. Knowledge discovery and data mining: towards a unifying framework. In: *KDD*. [S.l.: s.n.], 1996. v. 96, p. 82–88.

FREUND, Y.; MASON, L. The alternating decision tree learning algorithm. In: *icml*. [S.l.: s.n.], 1999. v. 99, p. 124–133.

GONÇALVES, E. J. et al. Uma abordagem baseada em agentes de apoio ao ensino a distância utilizando técnicas de engenharia de software. 2014.

HAN, J.; PEI, J.; KAMBER, M. *Data mining: concepts and techniques*. [S.l.]: Elsevier, 2011.

KAMPFF, A. J. C. Mineração de dados educacionais para geração de alertas em ambientes virtuais de aprendizagem como apoio à prática docente. 2009.

OLIVEIRA, R. B. T. d. *O processo de extração de conhecimento de base de dados apoiado por agentes de software.* Tese (Doutorado) — Universidade de São Paulo, 2000.

QUINLAN, J. R. *C4. 5: programs for machine learning*. [S.l.]: Elsevier, 2014.

RIGO, S. J.; CAZELLA, S. C.; CAMBRUZZI, W. Minerando dados educacionais com foco na evasão escolar: oportunidades, desafios e necessidades. In: *Anais do Workshop de Desafios da Computação Aplicada à Educação*. [S.l.: s.n.], 2012. p. 168–177.

RUSSEL, S. J.; NORVIG, P. Inteligência artificial: uma abordagem moderna. 2ª edição. *Rio de Janeiro, Brasil. Editora Campus*, 2004.

SILVA, S. B.; MACHADO, V. P.; ARAÚJO, F. N. Sistema tutor inteligente baseado em agentes na plataforma moodle para apoio as atividades pedagógicas da universidade aberta do piauí. In: *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*. [S.l.: s.n.], 2014. v. 3, n. 1, p. 592.

TAN, P.-N. et al. *Introduction to data mining*. [S.l.]: Pearson Education India, 2006.

WILGES, B. et al. Sistemas multiagentes: mapeando a evasão na educação a distância. *RENOTE*, v. 8, n. 1, 2010.