Utilizando Mineração de Dados e Sistemas Multiagentes na Análise da Evasão em Educação a Distância por meio do Perfil dos Alunos

Conference Paper · October 2016

CITATIONS
3

READS
4,034

4 authors:

Kaynan Coelho Lira

1 PUBLICATION **3** CITATIONS

SEE PROFILE

Marcos Antonio De Oliveira

Universidade Federal do Ceará

53 PUBLICATIONS **329** CITATIONS

SEE PROFILE

Enyo Gonçalves

Universidade Federal do Ceará

64 PUBLICATIONS **284** CITATIONS

SEE PROFILE

Regis Pires Magalhaes

Universidade Federal do Ceará

50 PUBLICATIONS **211** CITATIONS

SEE PROFILE

UTILIZANDO MINERAÇÃO DE DADOS E SISTEMAS MULTIAGENTES NA ANÁLISE DA EVASÃO EM EDUCAÇÃO A DISTÂNCIA POR MEIO DO PERFIL DOS ALUNOS

Kaynan C. Lira, Marcos A. de Oliveira, Regis P. Magalhães, Enyo José T. Gonçalves

[1]Sistemas de Informação – Universidade Federal do Ceará (UFC)
Av. José de Freitas Queiroz, 5003 – 63902-580 – Quixadá – Ce – Brasil

kaynancl.kc@gmail.com, {enyo,regismagalhaes,marcos.oliveira}@ufc.br

Resumo. *Sistemas Multiagentes (SMA) e Mineração de Dados (MD) vem sendo bastante utilizados na Educação a Distância (EAD), tendo como foco acompanhar o desempenho dos alunos visando o êxito escolar. Percebe-se que a evasão em EAD é consideravelmente alta e que constitui uma preocupação das instituições que oferecem cursos nesta modalidade. Foi desenvolvido um modulo para um SMA já existente, denominado SMAMoodle, tendo como finalidade acompanhar o comportamento dos alunos e identificar precocemente quando eles tendem a evasão. Foi possível identificar os padrões dos perfis dos alunos e auxiliar para mudança desse cenário.*

Abstract. *Multi-agent systems (MAS) and data mining (DM) has been widely used in distance education (DE), focusing on monitoring the performance of the students aiming at success in school performance. It is noticed that the dropout in DE is high, and that it is a concern of institutions which offer courses in such learning modality. This paper describes the development such learning modality already existing,named SMAMoodle, with the purpose to monitor the behavior of the students and to early identify when they tend to dropout. It was possible to identify the student profile patterns and help in changing that scenario.*

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. Um problema comum na EAD é a enorme quantidade de alunos que desistem ou possuem mau desempenho em seus cursos. De acordo com [Cavalcanti et al. 2014], 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, entre outros fatores.

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 sucesso dos alunos em seu desempenho escolar, 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].

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), 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 pela Universidade Estadual do Ceará (UECE), que tem como submodalidade da EAD a modalidade semipresencial, que possui no decorrer do curso algumas atividades presenciais. O sistema é composto por um conjunto de agentes, onde cada agente através de seus compromissos, se responsabilizam 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. Porém, ele não dar suporte a técnicas que permitam previamente evitar a evasão dos alunos.

Tendo em vista tal limitação, este trabalho propõe o desenvolvimento de um modulo módulo para identificação prévia de comportamentos que podem levar o aluno a evadir ou a ter mau desempenho no curso. Para que isso fosse possıvel, 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 possıvel prever o desempenho dos alunos para queo SMA forneça infirmações úteis para que o aluno possa ser ajudado. Com esses valores de desempenho identificados, informações são repassadas aos demais agentes do sistema para que decidam qual a melhor abordagem para mudar esse cenário 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, remediando a necessidade especifica de cada aluno, 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.

Este trabalho está dividido da seguinte forma: na Seção 2 é descrito o Referêncial Teórico que contempla os conceitos mais importantes para o desenvolvimento deste trabalho; na Seção 3 são descritos trabalhos relacionados e comparados com o presente trabalho; na Seção 4 são demonstrados os Experimentos utilizados para o desenvolvimento deste trabalho e os Resultados obtidos através deles; por fim na Seção 5 são descritas a conclusão e as ideias para os trabalhos futuros.

2. Referencial Teórico

Esta seção tem como princıpio 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.

[1] Disponível em: https://moodle.org

2.1. Evasão na Educação a Distância

A Educação a Distância (EAD) é o nome atribuıdo a uma modalidade de ensino que tem como caracterıstica principal o ensino e aprendizagem em que alunos e professores não necessitam estar juntos em um ambiente fısico como uma sala de aula durante a maior parte do tempo do curso [Cambruzzi 2014].

Muitos alunos encontram dificuldades em se adaptarem a EAD, isso acontece devido o costume do uso 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, um problema recorrente na EAD, relacionado ao elevado ındice 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].

2.2. Mineração de Dados Educacionais

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 uma classe é atribuída a cada instancia do conjunto de dados de treinamento a qual pertence [Júnior et al. 2015], e aprendizagem não supervisionada, que é quando as classes e as propriedades comuns entre as instancias ainda são desconhecida, sendo assim necessário observar os exemplos e reconhecer os padrões do modelo para que elas sejam definidas [Cambruzzi 2014].

Para o presente trabalho foram utilizadas duas técnicas:

1. Aprendizagem Supervisionada
   - Classificação: é utilizada para identificar modelos ou subgrupos de dados classificados de acordo com variáveis previamente definidas.
2. Aprendizagem Não Supervisionada:
   - Clusterização: é utilizada para identificar conjunto de categorias ou agrupamentos que possam descrever o comportamento dos dados selecionados.

De acordo com os trabalhos relacionados, os algoritmos que mais se adequam para a solução do problema em questão, são algoritmos de classificação e clusterização, por serem mais eficientes para prever ou descrever conjunto de dados de acordo com categorias nominais (evadido, aprovado, reprovado), os algoritmos elencados nos trabalhos relacionados para o problema correspondente foram:

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. ***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].

3. ***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. Dados valores de um conjunto de atributos, a técnica é capaz de navegar em uma árvore previamente criada poder realizar uma classificação. [Quinlan 2014].
4. ***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].

Foram realizados testes com esses algoritmos em relação ao modelo de dados aqui desenvolvido, tendo como fator comparativo a acurácia obtida com eles para que um fosse escolhido para ser utilizado para o presente trabalho. A Seção 4 descreve esse processo.

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].

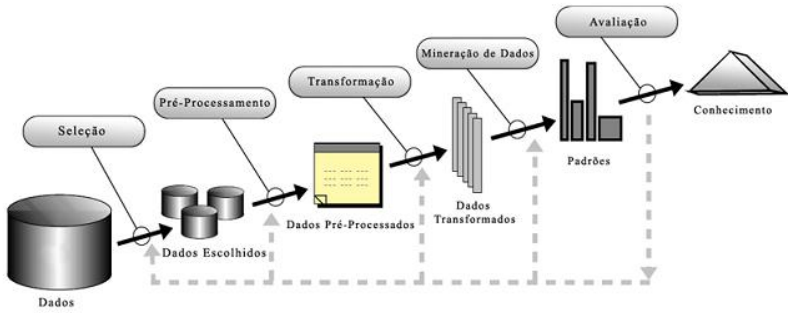

Figura 1. Processo de KDD

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. A seguir cada subseção irá explorar sucintamente cada uma das etapas do processo de KDD.

1. **Seleção**: Etapa em que será definidas a (s) fonte (s) que se relacionam com o domínio para a extração dos dados apropriados para o contexto.
2. **Pré-Processamento e Limpeza**: O subconjunto selecionado dos dados 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.
3. **Transformação dos Dados**: Etapa em que é necessário realizar a formatação e o armazenamento adequado dos dados.
4. **Mineração e Dados**: De acordo com [Tan et al. 2006], MD é uma forma de explorar e analisar dados de forma supervisionada ou não supervisionada, com o intuito de perceber padrões em grandes fontes de dados e assim obter informações relevantes para algum objetivo. Mais detalhes sobre MD já foram descritos na Seção 2.2.
5. **Interpretação e Avaliação**: Através dessa etapa é possível chegar à informação desejada, através da interpretação e avaliação dos padrões encontrados pela etapa de MD. Os usuários podem utilizar diversas ferramentas com funcionalidades estatísticas e de visualização para validarem ou julgarem um padrão irrelevante.

2.4. Sistemas Multiagentes

Sistemas Multiagentes podem ser entendidos como uma subárea da inteligência artificial, composta por agentes que, segundo [RUSSEL and 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[2](*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[3] (*Foundations of Intelligent Physical Agents*).

JADE possui uma extensão denominada JAMDER (*JADE to MAS-ML 2.0 Development Resource*) apresentada por [Lopes et al. 2012], ela possui suporte para implementação de SMA como organizações de agentes. Em organizações é onde são definidas as regras que são aplicadas aos agentes de uma determinada sociedade, em outras palavras, é o lugar onde é definido o que o agente desejará e realizará os seus comportamentos propostos.

Utilizando a metodologia multiagente será possível proporcionar dinamismo nas estratégias de ensino, levando em consideração as características individuais de aprendizagem de cada aluno. Os agentes modificam suas bases de conhecimento, percebem as intervenções do discente e são dotados da capacidade de aprender e adaptar suas estratégias de ensino mediante a interação com o aluno auxiliando o papel do tutor/professor.

3. Trabalhos Relacionados

Nesta seção são 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.

[2]Disponível em: http://jade.tilab.com
[3]Disponível em: http://www.fipa.org

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.

[Kampff 2009] utilizou dois algoritmos (*DecisionTree* e o *RuleLearner*) já descritos na Seção 2.2, e a ferramenta RapidMiner, que é uma ferramenta para pré-processamento e mineração de dados. O sistema de alertas desenvolvido por ele funciona através de geração de alertas definidos pelo professor (alertas fixos), e por alertas derivados da etapa de MD (alertas baseados em padrões).

Apesar do presente trabalho não ter como objetivo desenvolver um sistema de alertas, a metodologia utilizada por [Kampff 2009] para a identificação de alunos com mau desempenho foi de grande influencia.

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 de [Cavalcanti et al. 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.

Para a construção do modelo, [Cavalcanti et al. 2014] as notas dos alunos ao decorrer do semestre e para os testes realizados com o método de predição desenvolvido, foram utilizados os algoritmos *SimpleCart* e *J48*.

Apesar das influencias do trabalho de [Cavalcanti et al. 2014] sobre o processo de mineração de dados, o presente trabalho encapsulou todo o processo de mineração, predição e acompanhamento do aluno em Agentes desenvolvidos em *JADE*.

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

O trabalho de [Silva et al. 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* e para a descoberta de padrões nos dados obtidos através das interações dos usuários no *Moodle* foi utilizado o algoritmo *k-means*. Para que fosse possível classificar as novas instancias dos dados dos usuários em seus respectivos grupos, foi utilizado o algoritmo *J48*.

O presente trabalho assemelha-se bastante com o de Silva (silva2014sistema), 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. Apesar das diferenças o presente trabalho se influencia da abordagem de Silva (silva2014sistema) para a descoberta das classes que servirão para classificar os alunos.

4. Experimentos e Resultados

Esta seção descreve os procedimentos metodológicos que foram necessários para atingir os objetivos propostos neste trabalho.

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

Esse SMA 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.

Dentre os agentes que estão em funcionamento no sistema, o que é de maior utilidade para o contexto deste trabalho, é o Agente Companheiro de Aprendizagem. Ele é responsável por auxiliar o processo de aprendizagem dos alunos no decorrer do curso. De acordo com o desempenho do aluno, o agente o envia mensagens, estas mensagens podem conter informações de apoio, reforço ou sugestões de atividades. O seu comportamento é trivial para que alunos com mau desempenho possam melhorar.

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, ele 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. 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 as informações mais importantes que correspondessem às interações dos usuários na plataforma para a construção do modelo de dados em questão são:

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

4.2.2. Pré-Processamento

Para a construção do modelo, foram utilizados atributos que passaram por pré-processamento, visto que as informações contidas neles não estavam adequadas à aplicação das técnicas de mineração de dados e organizadas no banco de dados do *Moodle*. Para isso foi criado um *Data Mart* que é alimentado mensalmente. Ele também servirá para guardar o histórico de acompanhamento dos alunos no decorrer do período acadêmico.

Para a etapa de pré-processamento, foi desenvolvida uma aplicação na linguagem *Java*[4] utilizando *JDBC*[5] no ambiente de desenvolvimento *Eclipse*[6]. Essa aplicação mapeia o banco de dados do *Moodle* tendo como finalidade capturar as informações necessárias para inserir no *Data Mart*.

4.2.3. Organização e Fornecimento de Dados ao 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 neste 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. A seguir será descrito sucintamente cada uma das tabelas e suas finalidades.

4.2.4. Modelo, Mineração dos Dados e Descoberta de Padrões

Para a construção do Modelo foi utilizado a *API* fornecida pela ferramenta *Weka*[7], que é uma ferramenta para mineração de dados e 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, os dados são agrupados por aluno em um arquivo *arff*. Cada instancia do modelo de dados é composta pelos seguintes atributos:

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 et al. 2014], que se assemelha com o problema abordado neste trabalho. O *K-Means* é um algoritmo de clusterização, utiliza um parâmetro de

[4]Disponível em: https://www.java.com/pt_BR
[5]Disponível em: http://www.oracle.com/technetwork/java/javase/jdbc/index.html
[6]Disponível em: https://eclipse.org
[7]Disponível em: http://www.cs.waikato.ac.nz/ml/weka

Tabela 1. Descrição dos Dados que compõem o Modelo Inicial

Atributo	Descrição	Tipo
AVC	Número de Acessos ao Curso	Numérico
FAA	Número de Acessos ao Fórum	Numérico
PFA	Número de Postagens no Fórum	Numérico
AFA	Quantidade de Atividades Entregues	Numérico
FMD	Média das Notas das Atividades	Numérico
RAA	Número de Acessos aos Arquivos	Numérico
WAA	Número de Acessos às Wikis	Numérico

entrada k, que determina a quantidade de *clusters*, que é uma coleção de objetivos 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, e os *clusters* podem ter quantidade de elementos diferentes. Após o processo de clusterização no modelo de treinamento, os dados foram divididos em 5 grupos (MUITO BOM, BOM, REGULAR, RUIM e MUITO RUIM).

Com a descoberta dos clusters foi realizado um teste comparativo entre quatro 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 utilizado a técnica *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 et al. 2011]. Os dados foram divididos em 10 *folds*.

Tabela 2. 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.

4.2.5. 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.

4.2.6. 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 terá 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.

4.2.7. 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 contem 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 desenvolvido utilizando *JDBC* com o banco de dados *PostgreSQL*[8]. Para as interações do Agente Controlador de Evasão com o banco de dados do *Moodle*, foi aproveitado 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* [9], que é um *framework* que realiza o mapeamento objeto relacional (ORM), tendo como finalidade diminuir a complexidade do desenvolvimento da persistência dos dados.

[8]Disponível em: https://www.postgresql.org/
[9]Disponível em: http://hibernate.org/

4.2.8. Execução, Coleta de Dados e Testes

Para essa etapa o Agente Controlador de Evasão foi executado em um ambiente separado do SMA, afim de observar seu comportamento em relação a gerencia do processo de KDD nele encapsulado. Como já descrito na Seção 4.2.4, foi utilizado 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 4.2.4, utilizando o algoritmo *K-means*, afim 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%.

4.2.9. 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*, 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 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.

Dessa forma foi possível identificar com uma margem de acerto satisfatória os alunos que tendem a evasão, assim passando essas informações aos demais agentes do SMA, auxiliando aos alunos mudar esse cenário.

5. Conclusões e Trabalhos Futuros

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. 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*, o mais eficiente foi o *Random Forest*.

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 acom-

panhar 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.

Como trabalhos futuros podemos citar 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.

Referências

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. (1984). *Classification and regression trees*. CRC press.

Cambruzzi, W. L. (2014). Gvwise: uma aplicação de learning analytics para a redução da evasão na educação a distância.

Cavalcanti, Â. G. G., dos Santos, N. N., Cavalcanti, J. L., and Ramos, A. S. G. (2014). Mineração e visualização de dados educacionais: Identificação de fatores que afetam a motivação de alunos na educação a distância.

Cohen, W. W. (1995). Fast effective rule induction. In *Proceedings of the twelfth international conference on machine learning*, pages 115–123.

Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., et al. (1996). Knowledge discovery and data mining: towards a unifying framework. In *KDD*, volume 96, pages 82–88.

Gonçalves, E. J., de Oliveira, M. A., Junior, J. H. F., Feitosa, G. E., Mendes, D. H., Cortés, M. I., Feitosa, R. G., and Lopes, Y. S. (2014). Uma abordagem baseada em agentes de apoio ao ensino a distância utilizando técnicas de engenharia de software.

Han, J., Pei, J., and Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.

Júnior, M. S. R. F., Gonçalves, E. J. T., da Silva, T. L. C., and de Oliveira, M. A. (2015). Análise comportamental para proteção da criança nas redes sociais por meio de mineração de interações e sistemas multiagentes.

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

Lopes, Y. S., Gonçalves, E. J. T., Cortés, M. I., and de Campos, G. A. L. (2012). Jamder: Uma extensão do framework jade com foco em agentes.

Quinlan, J. R. (2014). *C4. 5: programs for machine learning*. Elsevier.

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

Silva, S. B., Machado, V. P., and Araújo, F. N. (2014). 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*, volume 3, page 592.

Tan, P.-N. et al. (2006). *Introduction to data mining*. Pearson Education India.