ARTIGO

Machine Learning Aplicado em Séries Temporais em um Sistema de Integração de Dados

Machine Learning Applied in Time Series in a Data Integration System

João Emmanuel D'Alkmin Neves[1]
José C. Conti[2]
Andréia R. Casare[3]

Submetido em: 28/05/2019 **Aceito em:** 26/07/2019 **Publicado em:** 20/08/2019

Resumo: *O processo de integração de dados é um método para extrair dados de diversas fontes, efetuar as devidas transformações, limpezas, normalizações e inserir os dados em tabelas. Esses dados são usados para processos decisórios pelos usuários em diversas áreas do conhecimento. Com o aumento da demanda por informações nos últimos anos, novas soluções estão sendo oferecidas a fim de tornar esse processo mais eficaz. No entanto, há escassez de processos que avaliem rotinas de processamento e as informações contidas nos logs dos processos de integração de dados. Nesse contexto, esse trabalho visa avaliar os dados contidos nas séries temporais desses processos aplicando a tarefa de agrupamento utilizando os algoritmos EM e K-means, que visa agrupar dados de acordo com seu grau de semelhança. Pretende-se com essa abordagem avaliar a eficácia das classes preexistes do processo de integração de dados, propor a criação de novas classes, além de apoiar especialistas no planejamento e dimensionamento de fluxos de processamento.*

Palavras-Chave: *agrupamento, integração de dados, séries temporais.*

Abstract*. The data integration process is a method for extract data from multiple sources, effect appropriate transformations, cleanups, normalizations and insert data into tables. This data is used for decision making by users in various areas of knowlegde. With the increasing demand for information in recent years, new solutions are being offered to make this process more effective. However, there is a dearth of processes that evaluate processing routines and the information contained in the data integration process logs. In this context, this work aims to evaluate the data contained in the time series of these processes by applying the grouping task using the EM and K-means algorithms, which aims to group data according to their degree of similarity. This approach aims to evaluate the effectiveness of pre-existing data integration process classes, propose the creation my new classes, and support experts in the planning and sizing of processing flows.*

Keywords: *grouping, data integration, time series.*

[1] Universidade Estadual de Campinas, Campinas (SP), Brasil. Email: jeneves@gmail.com
[2] Universidade Estadual de Campinas, Campinas (SP), Brasil. Email: conti30@gmail.com
[3] Universidade Estadual de Campinas, Campinas (SP), Brasil. Email: casareandreia@gmail.com

1. Introdução

O processo de integração de dados ocorre através de agendamento ou em tempo real, no qual há a extração de dados de diferentes fontes e formatos, tais como arquivos, outros bancos de dados, redes sociais, além de outras fontes. Esses dados passam pelo pré-processamento para efetuar a limpeza, integração, transformação e seleção dos dados, garantindo assim, a qualidade desses dados antes que sejam gravados em tabelas, *Data Warehouse* ou *Data Mart* específicos.

Normalmente os processos de extração, transformação e carga usam parâmetros de processamento, limpeza e mapeamento de dados, todavia, a maioria das abordagens de integração de dados não se concentram em analisar séries temporais desses processos, bem como não se utilizam de técnicas de mineração de dados.

Neste artigo, apresentamos a técnica de agrupamento usando os algoritmos *EM* e *K-means* para um Sistema de Integração de Dados para avaliar a precisão dessa tarefa, analisando os resultados na mineração de dados. Para isso, coletamos as séries temporais de processos de integração de dados de uma empresa do setor financeiro no período de 06/02/2019 a 21/02/2019 (durante 15 dias). Pode-se definir que série temporal é uma sequência de ações realizadas por uma ou mais variáveis em determinado intervalo de tempo, ocorrendo, normalmente, em intervalos uniformes (IMDADULLAH, 2014).

As informações armazenadas nas séries temporais dos registros foram: o tempo real expresso em segundos, a memória utilizada, a data de processamento e a classe a que pertencem os processos de carga. As saídas permitem observar e avaliar como os grupos estão distribuidos de acordo com suas características.

A contribuição deste trabalho é auxiliar pesquisadores e especialistas em integração de dados a planejar e dimensionar de forma mais precisa no sentido de manter seus processos mais eficientes através de técnicas de mineracao de dados, em especial a tarefa de agrupamento.

Este artigo está estruturado da seguinte forma: a Seção 2 apresenta trabalhos relacionados com integração de dados. A Seção 3 discute mineração de dados, séries temporais e tarefas de agrupamento. A Seção 4 apresenta a metodologia de nosso trabalho. A Seção 5 discute os resultados obtidos. A Seção 6 finaliza com conclusões e trabalhos futuros.

2. Trabalhos Relacionados

As tecnologias semânticas são usadas para produzir um conhecimento rico e significativo em torno da integração de dados e publicações *web*, é muito útil para uma série de aplicações inovadoras do *Big Data*. Para Bansal (2014) esta abordagem torna a integração de alguns conjuntos públicos de dados, a fim de criar ontologias de engenharia para uma boa compreensão de dados de diferentes fontes nas quais pode ser usado em várias aplicações.

Além disso, um processo de integração de dados é usado em vários campos, e seus *metadados* podem ser aplicados a todos os cenários de *Business Intelligence* (JAIN; GARG; SHARMA, 2015). O gerenciamento do processo de extração é otimizado usando operadores que ajudam na redução de sua complexidade. Alguns operadores são projetados para entender facilmente cada camada de extração e desempenham um papel vital na área de *Business Intelligence* Esses operadores, *Create*, *Reverse*, *Join* e *Undo* foram usados nesta abordagem para ajudar a reduzir sua complexidade. Em cada camada de aplicação, os metadados são organizados para monitorar o fluxo de dados dentro do ambiente através do conceito de lista associado a cada camada da arquitetura e é possível controlar alguns processos de Integração de Dados. Na camada de operação são construídos modelos de processo usando ferramentas de modelagem, salvando esses modelos em bancos de dados para processar sua execução ou simulação.

Outro requisito no processo de integração de dados diz respeito à qualidade dos dados em que é reconhecido como um dos problemas de gerenciamento de dados mais importante. As inconsistências ocorrem quando diferentes versões dos mesmos dados aparecem em sistemas diferentes (KALE; APARADH, 2016). Para melhorar esta situação, chaves são usadas para identificar objetos, isto significa que atributos ou um conjunto de atributos buscam seus objetos e se comparam com os objetos de origem. Se houver a inconsistência desses atributos, ele é resolvido para selecionar o objeto mais adequado, desde que seja a melhor fonte de dados. É usado neste trabalho, lógica triangular difusa para representar os valores de critérios qualitativos para cada atributo.

Lenzerini (2002) concentra sua pesquisa abordando alguns problemas nos sistemas de integração de dados, é uma situação importante em aplicações do mundo real e essa abordagem é caracterizada por uma série de requisitos que são interessantes do ponto de vista teórico. O autor apresenta uma visão geral do material a ser divulgado em um tutorial sobre integração de dados. Este tutorial está centrado em alguns dos problemas teóricos relevantes neste processo. Neste cenário, os principais aspectos são discutidos: Modelando um aplicativo na integração de dados, processamento de consultas, fontes de dados inconsistentes para análise e raciocínio na forma como as consultas são preparadas. O autor sugere que outros aspectos também sejam discutidos no contexto do processo de integração de dados, incluindo a qualidade e a limpeza dos dados em uma estrutura de integração de dados, bem como a otimização e avaliação de consultas postadas no sistema.

Prema e colaboradores (2013) propõem um método conhecido como *Hyper ETL* com o objetivo de reduzir o tempo de processamento, implementação, custo de manutenção, aumentar o desempenho e a confiabilidade e ainda otimizar o gerenciamento de metadados. Este trabalho analisa os problemas encontrados nas ferramentas existentes e compara os parâmetros do *Hyper ETL* com o *ETL* existente. Esta ferramenta amplia o método de agregação, transmite informações de forma inteligente e é útil para a tomada de decisões eficazes. Alguns dos parâmetros utilizados na avaliação foram: escalabilidade, gerenciamento, utilização da *CPU*, velocidade e consistência. Os resultados indicam que tanto o processo *ETL* tradicional, como o *Hyper ETL*, oferecem o mesmo nível de gerenciamento, rastreabilidade, mobilidade e

consistência. Além disso, o *Hyper ETL* é um parâmetro ao avaliar o custo da manutenção.

Para Elgendy e Elragal (2014) devido ao rápido crescimento de *Big Data*, faz-se necessário o estudo de soluções para manipular e extrair valor e conhecimento desses conjuntos de dados. Além disso, os tomadores de decisão precisam ser capazes de obter informações valiosas a partir desses dados e mudar rapidamente sua análise de transações diárias para a rede social com dados de interações do cliente, também conhecido como dados não estruturados. Os autores avaliam alguns dos diferentes métodos de análise e ferramentas que podem ser aplicados a grandes volumes de dados e oferecem oportunidades em muitas áreas do conhecimento.

3. Referencial Teórico

3.1. Mineração de dados

A mineração de dados é um campo de pesquisa que engloba diversas áreas, tais como: inteligência artificial, bancos de dados, redes neurais, aprendizagem de máquina, estatística, sistemas baseados em conhecimento, recuperação da informação, visualização de dados e computação de alto desempenho (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996). Atualmente, o volume de dados gerado cresce ano a ano. Esses dados, não-estruturados, compõem cerca de 90% do universo digital. Todavia, mais dados não significa mais conhecimento. A mineração de dados possibilita filtrar o ruído existente nos dados digitais, buscar o que é relevante e, então, gerar informação a partir dos resultados (BROWN, 2012).

3.2. Séries Temporais

A análise de séries temporais, no processo de mineração de dados, tornou-se uma fonte de investigação importante em diversas áreas, conforme observado nos artigos de Andrienko *et al.* (2010), Otranto (2010) e Mezer *et al.* (2009).

Uma série temporal pode ser considerada como sendo uma coletânea de valores que são obtidos através de medições sequenciais ao longo de um determinado tempo (ESLING, et al., 2012). Os estudiosos utilizam algoritmos de mineração complexos para realizar pesquisas, sendo que as principais etapas são: classificação, regressão, agrupamento e predição.

Brockwell e Davis (1997) definem as séries temporais como sequências de observações organizadas no espaço ou no tempo. As medições obtidas se alteram com passar do tempo e, essa variação é definida como sendo uma série temporal.

Uma série temporal Z é representada matematicamente por: $Z = \{Z_{t-1}, Z_{t-2}, Z_{t-3}...\}$, ou seja, a Série Temporal Z corresponde a um conjunto de medições relacionadas ao tempo t. Há dois tipos de séries temporais: a) contínuas, onde as observações estão em todos os instantes de tempo, como por exemplo um eletrocardiograma; e b)

discretas, onde existe a observação de um espaço de tempo regular, como exemplo o log de servidor.

A análise das séries temporais baseia-se no uso de modelos matemáticos e estatísticos objetivando compreender e quantificar os fenômenos de variação temporal. Essa análise ocorre através da seleção de dados passados para realizar a predição de dados futuros, visando através dessa análise, construir modelos que permitem prever a série temporal futura (OLIVEIRA, 2007).

3.3. Tarefa de Agrupamento

A tarefa de agrupamento ou clusterização pode ser definida como um método de classificação não supervisionada de padrões (observações, itens de dados ou vetores de recursos) em grupos (clusters) com base na utilização de algoritmos. Dois algoritmos amplamente utilizados são o *K-means* e o *Expectation Maximization* (*EM*). *EM* e *K-means* são semelhantes no sentido de que eles permitem o refinamento do modelo de um processo iterativo para encontrar o melhor agrupamento. No entanto, o *K-means* utiliza uma medida de distância que calcula a distância entre cada um dos itens de dados; por outro lado o *EM* usa métodos estatísticos (JUNG et al.; 2014).

O problema de agrupamento foi abordado em muitos contextos e por pesquisadores em diversas disciplinas; Isso reflete seu amplo apelo e utilidade como uma das etapas da análise de dados exploratórios (JAIN et al.; 1999).

4. Metodologia

A metodologia utilizada para realizar esse trabalho seguiu as seguintes etapas: i) coletado séries temporais de processos de integração de dados de 06/02/2019 a 21/02/2019 durante 15 dias, contendo os seguintes dados: tempo real, a memória utilizada, a data de processamento dos logs, a classe a qual cada Job pertence (Tabela 1); ii) realizar o pré-processamento dos dados coletados; iii) escolha de uma ferramenta de mineração de dados para realizar a tarefa de agrupamento; iv) escolha de dois algoritmos de agrupamento; v) realizar a clusterização (agrupamento) utilizando conjuntos de (treinamento e teste); vi) análise e avaliação dos clusters gerados pelos dois algoritmos de agrupamento escolhido.

Na etapa (ii) a base de dados foi preparada para ser utilizada na tarefa de agrupamento. A base original possuía um atributo classe, esse atributo classe representa a categoria dos logs de integração de dados, que são determinados de acordo com algumas premissas das quais seguem uma ordem de acordo com os atributos relacionados a chegada de arquivos para processamento, bem como as dependências dos próximos Jobs em relação aos seus predecessores. Diante disso, excluímos o atributo classe, pois a tarefa de mineração utilizada nesse trabalho foi o agrupamento.

A ferramenta escolhida na etapa (iii) foi a *Waikato Environment for Knowledge Analysis* (WEKA) na versão 3.8.1. A escolha se deu devido a ferramenta ser amplamente utilizada para tarefas de mineração de dados, pois é de uso livre e possui

uma série de algoritmos para as tarefas de mineração. Fornece as funcionalidades para pré-processamento, classificação, regressão, agrupamento, regras de associação e visualização.

Na etapa (iv) foram escolhidos os algoritmos *Expectation Maximization (EM*) e *K-means*. O algoritmo *EM* é aplicado em situações onde se deseja estimar um conjunto de parâmetros que descreve uma distribuição de probabilidade, usando dados estatísticos e o *K-means* utiliza uma medida de distância na tarefa de agrupamento, no nosso trabalho foi utiliza a Distância Euclidiana.

Já na etapa (v) foi realizada a tarefa de agrupamento, sendo executada da seguinte maneira: primeiramente 100 % dos dados para o conjunto de treinamento e na sequência 80% para treinamento e 20% para teste e por último a avaliação das classes para os clusters gerados pelos algoritmos.

Tabela 1: Volume de processamento de cada classe

	Dados reais	
Classe	#	%
1	1123	73,78%
2	71	4,66%
3	24	1,58%
4	52	3,42%
5	203	13,34%
6	12	0,79%
8	18	1,18%
9	4	0,26%
10	4	0,26%
11	4	0,26%
99	7	0,46%
Total	**1522**	**100,00%**

Fonte: Elaboração própria (2017)

5. Resultados e Discussão

Os resultados obtidos na tarefa de agrupamento utilizando o algoritmo *EM* foram os seguintes: Para 100% do processamento na tarefa de agrupamento, para o conjunto de treinamento, o algoritmo selecionou 13 grupos (*clusters*) das 1522 amostras. Nessa distribuição, observa-se uma maior concentração no grupo 4 contendo 660 observações que corresponde a 43,36% do total de amostras, conforme apresentado na Tabela 2.

Tabela 2: Conjunto de treinamento - *EM*

Treinamento		
Cluster	#	%
0	67	4,40%
1	8	0,53%
2	158	10,38%
3	5	0,33%
4	660	43,36%
5	175	11,50%
6	7	0,46%
7	4	0,26%
8	141	9,26%
9	17	1,12%
10	58	3,81%
11	14	0,92%
12	208	13,67%
Total	**1522**	**100,00%**

Fonte: Elaboração própria (2017)

Quando selecionamos 20% da base para o conjunto de teste e aplicamos a tarefa de agrupamento, o algoritmo apresenta 05 grupos de 305 amostras, sendo que a maior concentração acontece no grupo 2 com 134 observações que representa 43,93% do conjunto de teste que contém 305 amostras, conforme observa-se na Tabela 3.

Tabela 3: Conjunto de teste - *EM*

Teste		
Cluster	#	%
0	20	6,56%
1	85	27,87%
2	134	43,93%
3	26	8,52%
4	40	13,11%
Total	**305**	**100,00%**

Fonte: Elaboração própria (2017)

Na avaliação dos resultados da classificação utilizando o algoritmo *EM,* observa-se que foram identificadas classes para os *clusters* 2, 4, 5, 8, 10 e 12. Por outro lado, não foram identificadas classes para os demais *clusters*, conforme apresentado na Tabela 4.

Tabela 4: Resultados da avaliação - *EM*

	Classe										
Cluster	1	2	3	4	5	6	8	9	10	11	99
0											
1											
2											
3											
4											
5											
6											
7											
8											
9											
10											
11											
12											

Fonte: Elaboração própria (2017)

Na classificação das instâncias, identifica-se que o algoritmo *EM* classificou 50,27% corretamente e 49,73% incorretamente, conforme apresentado na Tabela 5.

Tabela 5: Classificação das instâncias - *EM*

Classificação das instâncias	%
Instâncias classificadas corretamente	50,27
Instâncias classificadas incorretamente	49,73

Fonte: Elaboração própria (2017)

Os resultados obtidos na tarefa de agrupamento com as séries temporais de sistemas de integração de dados usando o algoritmo *K-means* foram os seguintes: Para 100% do processamento na tarefa de agrupamento, para o conjunto de treinamento, utilizamos 13 grupos (clusters) das 1522 amostras para manter a mesma quantidade de *clusters* como resultado do algoritmo *EM*. Nessa distribuição, observa-se uma maior

concentração no grupo 9 contendo 173 observações que corresponde a 11,37% do total de amostras, conforme apresentado na Tabela 6.

Tabela 6: Conjunto de treinamento – *K-means*

Treinamento		
Cluster	**#**	**%**
0	129	8,48%
1	145	9,53%
2	33	2,17%
3	128	8,41%
4	132	8,67%
5	127	8,34%
6	128	8,41%
7	129	8,48%
8	137	9,00%
9	173	11,37%
10	61	4,01%
11	73	4,80%
12	127	8,34%
Total	**1522**	**100,00%**

Fonte: Elaboração própria (2017)

Quando selecionamos 20% da base para o conjunto de teste e aplicamos a tarefa de agrupamento, o algoritmo *K-means* apresenta 05 grupos de 305 amostras para manter a mesma quantidade de clusters como resultado do algoritmo *EM*, sendo que a maior concentração acontece no grupo 1 com 147 observações que representa 48,20% do conjunto de teste, conforme observa-se na Tabela 7.

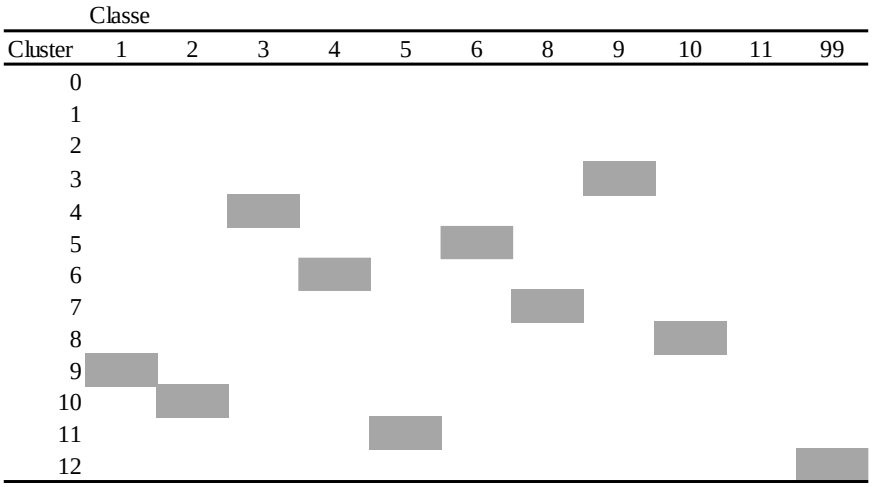
Tabela 7: Conjunto de teste – *K-means*

Teste		
Cluster	**#**	**%**
0	55	18,03%
1	147	48,20%
2	42	13,77%
3	26	8,52%
4	35	11,48%
Total	**305**	**100,00%**

Fonte: Elaboração própria (2017)

Na avaliação dos resultados da classificação utilizando o algoritmo *K-means,* observa-se que foram identificadas classes para os *clusters* 3, 4, 5, 6, 7, 8, 9, 10, 11 e 12. Por outro lado, não foram identificadas classes para os demais *clusters*, conforme apresentado na tabela 8.

Tabela 8: Resultados da avaliação – *K-means*



Fonte: Elaboração própria (2017)

Na classificação das instâncias, identifica-se que o algoritmo *K-means* classificou 13,54% corretamente e 86,46% incorretamente, conforme apresentado na Tabela 9.

Tabela 9: Classificação das instâncias – *K-means*

Classificação das instâncias	%
Instâncias classificadas corretamente	13,54
Instâncias classificadas incorretamente	86,46

Fonte: Elaboração própria (2017)

Analisando os resultados de agrupamentos obtidos pelos algoritmos *EM* e *K-means*, pode-se inferir que o algoritmo *EM* se saiu melhor na tarefa de agrupamento, pois conseguiu classificar corretamente 50,27% instâncias enquanto que o algoritmo *K-means* classificou apenas 13,54% instâncias corretamente. Adicionalmente, percebe-se que para a avaliação das instâncias, o algoritmo *K-means* mostrou-se mais rápido em

relação ao algoritmo *EM* para avaliação do conjunto dos dados das classes para a formação dos grupos.

6. Conclusão

Este artigo apresentou conceitos e características de mineração de dados e tarefa de agrupamento utilizando os algoritmos *EM* e *K-means* com o objetivo de encontrar informações e buscar agrupamentos em séries temporais baseada em dados de logs de processamento de sistemas de integração de dados.

Em especial, o presente estudo investigou a possibilidade da aplicação de técnicas de mineração de dados a fim de realizar inferências relativas ao desempenho do processamento de sistemas de integração de dados. Essas inferências consistiram em verificar quais os melhores agrupamentos para as séries temporais de acordo com os dados dos logs e, efetuar uma comparação com os grupos já existentes, buscando a criação dos melhores agrupamentos.

Os resultados obtidos com os experimentos apontam para viabilidade de se realizar inferências relativas aos melhores grupos. Em virtude disso, os algoritmos de agrupamento utilizados, *EM* e *K-means* se mostraram eficientes em relação a formar grupos de séries temporais. Nesse contexto, foi possível verificar que determinadas séries foram agrupadas anteriormente em categorias que apresentavam variações de tempo de execução e memória utilizada, conforme pode ser visto na Tabela 1. Por este fato, conclui-se que os agrupamentos anteriores apresentavam falhas.

Não é possível afirmar o que, de fato, provocou as eventuais falhas. As mesmas podem ter sido geradas por erro de classificação, dados com ruídos, ou ainda por defeitos ocorridos durante a categorização. O essencial é que uma vez encontrado a existência de falhas, fica a critério dos pesquisadores e profissionais da área a execução de providencias para corrigir e solucionar os problemas a fim de garantir os resultados das pesquisas.

As estimativas sobre o tempo real de execução, a memória utilizada e a data de processamento, disponibilizadas durante a realização do processo, são úteis para pesquisadores e profissionais da área de integração de dados dimensionarem os melhores requisitos para sistemas, equipamentos e previsão de tempo e desenvolverem estratégias que busquem maximizar os melhores resultados, como ganho computacional e redução do tempo de execução das tarefas. Além disso, essas informações disponibilizadas continuamente nos logs podem auxiliar no desenvolvimento de ações com os processos em andamento e não somente para planejamentos futuros.

Diante disso, esse trabalho pode potencializar o uso de tarefas de mineração de dados em sistemas de integração, além de apoiar especialistas no planejamento e dimensionamento de fluxos de processamento de *ETL*.

Como proposta para trabalhos futuros, lançamos alguns desafios que podem direcionar ao uso de outros algoritmos de agrupamento, utilizar tarefa de classificação,

testar outras distribuições no conjunto de teste e avaliar as séries temporais em logs de processos de integração em outro período de tempo.

Referências Bibliográficas

ANDRIENKO, G.; ANDRIENKO, N.; MLADENOV, M.; MOCK, M.; POELITZ, C. **Extracting Events from Spatial Time Series**. IEEE 14th International Conference on Information Visualisation, p. 48-53, 2010.

BANSAL, S. K. **Towards a Semantic Extract-Transform-Load (ETL) framework for Big Data Integration**. 2014 IEEE International Congress on Big Data, 2014.

BROCKWELL, P. J.; DAVIS, R. A. **Introduction to Time Series and Forecasting**. 2nd Edition, Springer Texts in Statistics, 1997

BROWN, M. **Data Mining as a Process**. IBM Developer Works Library. December 11, 2012.

ELGENDY, N.; ELRAGAL, A. **Big Data Analytics**: A Literature Review Paper. P. Perner (Ed.): ICDM 2014, LNAI 8557, pp. 214–227, 2014. Springer International Publishing Switzerland, 2014.

ESLING, P.; AGON, C.; RECHERCHE, I. D. **Time-Series Data Mining**. ACM Computing Surveys, v. 45, n. 1, 2012.

FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. **From Data Mining to Knowledge Discovery in Databases**. Artificial Intelligence Magazine, v. 17, n. 3, 1996.

IMDADULLAH , M. **Time Series Analysis and Forecasting Time Series**. Basic Statistics and Data Analysis. January 2014.

JAIN, A.; GARG, S.; SHARMA, N. **The Management of Conformed ETL Architecture**. International Journal of Computer Apllications (0975-8887), Volume 118 - No. 10, May 2015.

JAIN, A. K.; MURTY, M. N.; FLYNN, P. J. **Data Clustering**: A Review. ACM Computing Surveys. 1999.

JUNG, Y. G.; KANG, M. S.; HEO, J. **Clustering performance comparison using K-means and expectation maximization algorithms**. Biotechnology & Biotechnological Equipment, 2014. Vol. 28, No. S1, S44_S48, http://dx.doi.org/10.1080/13102818.2014.949045.

KALE, D. R.; APARADH, S. Y. **A Study of a Detection and Elimination of Data Inconsistency in Data Integration**. IJSRSET1621111 | Received: 20 February 2016 | Accepted: 03 March 2016 | January-February 2016 [(2)1: 532-535], 2016.

LENZERINI, M. **Data Integration: A Theoretical Perspective**. ACM PODS 2002, June 3-6. Madison, Wisconsin, USA, 2002.

MEZER, A.; YOVEL, Y.; PASTERNACK, O.; GORNE, T.; ASSAF , Y. **Cluster analysis of resting-state fmri time series**. Neuroimage, v. 45, n. 4, p. 1117–1125, 2009.

OLIVEIRA, P. C. **Séries Temporais**: Analisar o Passado, Predizer o Futuro. Analysis, p. 3-6, 2007.

OTRANTO, E. **Identifying Financial Time Series with Similar Dynamic Conditional Correlation**. Computational Statistics & Data Analysis 54, 115, 2010.

PREMA, A.; SUJATHA, N.; PETHALAKSHMI, A. **A Comparative analysis of ETL and Hyper ETL**. International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), Volume 2, Isse 6, November - December 2013, 2013.