

TEMAR: Uma Ferramenta para Análise Temporal de Regras de Associação em Dados de Repositórios de Software

Silvana de Andrade Gonçalves
Instituto Federal do Acre
Acre, Brasil
silvana.goncalves@ifac.edu.br

Alexandre Plastino
Instituto de Computação,
Universidade Federal Fluminense
Rio de Janeiro, Brasil
plastino@ic.uff.br

Daricélio Moreira Soares
Universidade Federal do Acre
Acre, Brasil
daricelio.soares@ufac.br

Resumo

A análise de dados provenientes de projetos de software tem sido amplamente utilizada para investigar padrões relacionados ao desenvolvimento distribuído. Nesse contexto, regras de associação permitem identificar relações frequentes entre atributos presentes em bases de dados de projetos de software, como informações sobre *pull requests* e contribuidores. Entretanto, muitos estudos consideram apenas a extração das regras na base completa, desconhecendo mudanças nos padrões ao longo do tempo. Este artigo apresenta a TEMAR, uma ferramenta para apoio à análise temporal de regras de associação em dados de projetos de software. A proposta permite carregar bases de dados em formato CSV oriundas de repositórios de software, aplicar diferentes estratégias de particionamento temporal, minerar regras de associação e visualizar a evolução das medidas de Suporte, Confiança e *Lift* por meio de gráficos. A ferramenta também auxilia na interpretação da evolução temporal dos padrões encontrados em projetos de código aberto. Por fim, um cenário de demonstração utilizando dados reais de projetos hospedados no GitHub é apresentado para ilustrar suas funcionalidades.

Vídeo de demonstração: <https://doi.org/10.5281/zenodo.20367413>

Palavras-chave

regras de associação, análise temporal, mineração de repositórios de software, mineração de dados, visualização de dados.

1 Introdução

Projetos de código aberto geram continuamente grandes volumes de dados relacionados ao processo de desenvolvimento colaborativo, incluindo informações sobre *commits*, *issues* e *pull requests*. Esses dados têm sido amplamente explorados em estudos de Engenharia de Software e Mineração de Repositórios de Software (Mining Software Repositories – MSR).

Neste contexto, regras de associação constituem uma técnica de mineração de dados utilizada para identificar relações frequentes entre atributos de uma base. Em estudos com *pull requests*, essas regras permitem investigar relações entre características das contribuições e seus resultados, identificando padrões relacionados à aceitação [10], rejeição [11] e comportamento de contribuidores em projetos de software [9][12].

Entretanto, muitos estudos extraem regras considerando apenas a base completa [9][10][11][12], desconhecendo possíveis mudanças temporais nos padrões encontrados. Em ambientes colaborativos, alterações em diretrizes de contribuição e práticas de revisão podem modificar medidas como Suporte, Confiança e *Lift*.

De fato, a introdução do arquivo *contributing.md* pode influenciar padrões relacionados à aceitação de *pull requests* de novatos [6].

Apesar da relevância desse tipo de análise, não foram identificadas ferramentas que apoiassem a análise temporal de regras de associação em dados de Engenharia de Software. Ferramentas amplamente utilizadas para mineração de dados, como Weka [7], Orange [4], Knime [2] e a Linguagem R [8] permitem a extração de regras de associação a partir de uma base de dados, porém não oferecem, de forma integrada, mecanismos para particionamento temporal da base, comparação das medidas das regras entre diferentes períodos e visualizações voltadas à análise de sua evolução temporal, exigindo implementações adicionais por parte do usuário.

Diante desse cenário, este artigo apresenta a TemAR, uma ferramenta desenvolvida em Python¹ e Streamlit² para apoiar a análise temporal de regras de associação extraídas de dados de projetos de software. A ferramenta permite carregar bases de dados em formato CSV oriundas de repositório de software, particionar os dados temporalmente por diferentes estratégias, minerar regras de associação e comparar visualmente a evolução temporal das medidas de interesse das regras.

Como principais contribuições, a ferramenta oferece: (i) suporte à extração e análise temporal de regras de associação em dados de Engenharia de Software; (ii) diferentes estratégias de particionamento temporal; (iii) visualizações gráficas para comparação temporal das medidas de interesse; e (iv) apoio à interpretação da evolução temporal dos padrões em projetos de código aberto.

O restante deste artigo está organizado da seguinte forma. A Seção 2 apresenta a fundamentação teórica relacionada às regras de associação e sua análise temporal. A Seção 3 descreve a visão geral da ferramenta, incluindo arquitetura, fluxo de execução e tecnologias utilizadas. A Seção 4 apresenta um cenário de demonstração utilizando projetos reais de código aberto. Por fim, a Seção 5 apresenta as conclusões e trabalhos futuros.

2 Fundamentação Teórica

A ferramenta proposta apoia-se em conceitos relacionados à mineração de dados e à análise temporal de padrões extraídos de repositórios de software. Em particular, utiliza regras de associação para identificar relações frequentes entre características presentes em dados de *pull requests* e explora a dimensão temporal dessas regras com o objetivo de investigar mudanças nos padrões encontrados ao longo do tempo. A seguir, são apresentados os principais conceitos relacionados à abordagem proposta.

¹<https://www.python.org>

²<https://streamlit.io/>

2.1 Regras de Associação Multidimensionais

Regras de associação são utilizadas para identificar relações frequentes entre atributos presentes em uma base de dados. No contexto de Engenharia de Software, essas regras podem auxiliar na descoberta de padrões relacionados ao desenvolvimento, revisão e aceitação de contribuições em projetos de código aberto [9][12].

Em bases de dados relacionais, é comum minerar regras de associação multidimensionais, nas quais os itens da regra representam condições associadas a diferentes atributos da base. Formalmente, uma regra pode ser representada na forma $X \rightarrow Y$, onde X representa o antecedente da regra e Y representa o consequente. Tanto X quanto Y são conjunções de condições sobre atributos distintos.

Como exemplo, considere a regra: $\text{FirstPull} = \text{"true"} \rightarrow \text{Status} = \text{"accepted"}$, que indica que *pull requests* submetidos por novos contribuidores tendem a ser aceitos com determinada frequência na base analisada.

A qualidade e relevância das regras de associação são avaliadas por suas medidas de interesse, sendo as mais utilizadas o Suporte, a Confiança e o *Lift*.

O Suporte da regra representa a frequência com que antecedente e consequente ocorrem simultaneamente na base de dados [1]. Essa medida indica o quanto uma regra é representativa dentro do conjunto analisado, sendo calculada pela Equação 1.

$$\text{Sup}(X \rightarrow Y) = \frac{T(X \cup Y)}{T} \quad (1)$$

Nesta equação, $T(X \cup Y)$ representa a quantidade de registros que satisfazem simultaneamente as condições do antecedente e do consequente, enquanto T representa o total de registros da base.

A Confiança mede a probabilidade de ocorrência do consequente Y dado que o antecedente X ocorreu [1]. Essa medida indica a fração de registros que satisfazem as condições do antecedente e do consequente entre aqueles que satisfazem as condições do antecedente, conforme apresentado na Equação 2.

$$\text{Conf}(X \rightarrow Y) = \frac{T(X \cup Y)}{T(X)} \quad (2)$$

Nesta formulação, $T(X)$ corresponde ao número de registros que satisfazem as condições do antecedente da regra.

A medida *Lift* avalia o grau de dependência entre antecedente e consequente, comparando a Confiança observada da regra com a frequência de ocorrência do consequente [1]. Essa medida é calculada pela Equação 3.

$$\text{Lift}(X \rightarrow Y) = \frac{\text{Conf}(X \rightarrow Y)}{\text{Sup}(Y)} \quad (3)$$

Valores de *Lift* maiores que 1 indicam dependência positiva entre X e Y , sugerindo que a ocorrência do antecedente aumenta a probabilidade de ocorrência do consequente. Valores menores que 1 indicam dependência negativa, enquanto valores iguais a 1 sugerem independência entre os elementos da regra.

Neste trabalho, as regras de associação são extraídas de bases contendo informações sobre *pull requests* de projetos de código aberto hospedados no GitHub³. Os registros analisados incluem características relacionadas aos contribuidores, aos *pull requests* e ao

processo de revisão, permitindo investigar padrões de contribuição e suas variações ao longo do tempo.

2.2 Análise Temporal de Regras de Associação

A análise de regras de associação normalmente é realizada considerando uma visão estática da base de dados, na qual as regras são extraídas a partir de um conjunto completo de registros [9][12]. Embora essa abordagem permita identificar padrões frequentes, ela pode ocultar mudanças relevantes que ocorrem ao longo do tempo, dificultando a percepção da evolução dos comportamentos observados nos dados.

Em projetos de software, padrões de contribuição podem mudar devido à adoção de novas ferramentas, alterações em diretrizes ou evolução do projeto [6]. Nesse contexto, a análise temporal de regras de associação permite investigar o comportamento desses padrões em diferentes períodos [5].

Uma estratégia adotada em estudos de análise temporal consiste no particionamento da base de dados em subconjuntos correspondentes a diferentes períodos de tempo [3][6][13]. A partir dessas partições, as regras de associação podem ser mineradas e comparadas por meio de suas medidas de interesse, permitindo identificar variações temporais em padrões previamente observados.

Entretanto, esse processo geralmente envolve etapas manuais de preparação dos dados, particionamento temporal, extração das regras e comparação das medidas obtidas em diferentes períodos. Nesse contexto, a ferramenta proposta busca integrar essas atividades em um único ambiente, oferecendo suporte ao particionamento temporal, mineração das regras de associação e visualização gráfica da evolução temporal das medidas de interesse.

Na ferramenta apresentada neste estudo, a análise temporal é realizada por meio da visualização e comparação das medidas das regras extraídas da base completa e das diferentes partições temporais dos dados.

3 Visão Geral da Ferramenta

A ferramenta proposta foi desenvolvida com o objetivo de apoiar a análise temporal de regras de associação extraídas de dados de repositórios de software. Sua utilização permite que desenvolvedores, gerentes de projetos e pesquisadores realizem desde o carregamento da base de dados até a visualização gráfica das variações temporais das regras mineradas.

A ferramenta foi implementada como uma aplicação web, utilizando Python e Streamlit, permitindo a execução das análises por meio de uma interface interativa e acessível via navegador. O processo de mineração das regras de associação é realizado utilizando a biblioteca MLxtend⁴, que fornece implementação do algoritmo *Apriori* [1] e funções auxiliares para a geração das regras e o cálculo das medidas de interesse. O código fonte da ferramenta é disponibilizado publicamente sob licença MIT.

3.1 Arquitetura da Ferramenta

A arquitetura da ferramenta é composta por módulos responsáveis pelo processamento e análise temporal das regras de associação,

³<https://github.com>

⁴<https://rasbt.github.io/mlxtend/>

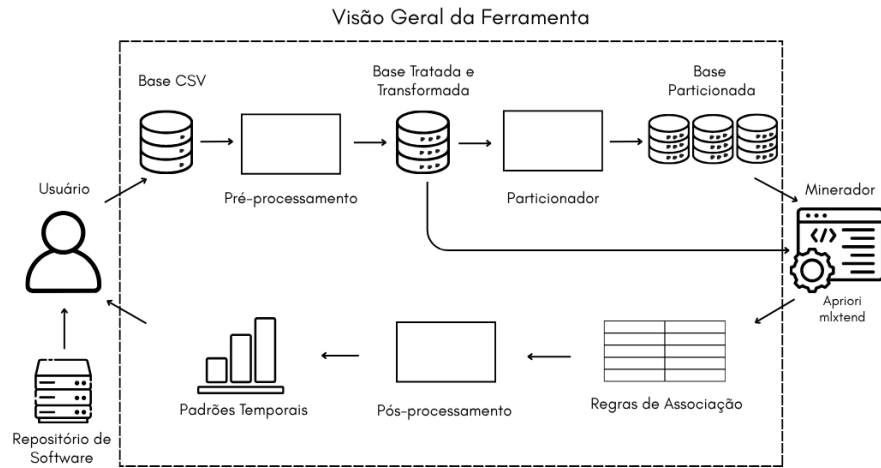


Figura 1: Arquitetura da Ferramenta de Análise Temporal de Regras de Associação.

conforme ilustrado na Figura 1. Os principais módulos são: pré-processamento, particionamento temporal, mineração de regras e pós-processamento.

O módulo de pré-processamento é responsável pela importação da base de dados em formato CSV, validação das informações necessárias para a análise e preparação dos dados para a mineração das regras de associação. Nessa etapa, os atributos categóricos são organizados para utilização pelo algoritmo *Apriori*, enquanto atributos do tipo data são temporariamente desconsiderados e posteriormente utilizados no particionamento temporal.

A ferramenta utiliza como entrada um arquivo CSV contendo atributos categóricos e uma única coluna temporal, identificada automaticamente para o particionamento dos dados. Não são realizadas etapas de preparação dos dados, como discretização de atributos numéricos, tratamento de valores ausentes ou consolidação de múltiplas colunas temporais. Eventuais atributos numéricos presentes na base são automaticamente ignorados durante a mineração, sendo responsabilidade do usuário fornecer uma base previamente preparada para a mineração.

O módulo de particionamento é responsável pela divisão da base em subconjuntos temporais utilizados durante a análise. Atualmente, a ferramenta suporta três estratégias de particionamento: definição manual de marcos temporais, intervalos temporais de mesmo tamanho e particionamento baseado em quantidades equivalentes de registros.

Após o pré-processamento e particionamento, os dados são encaminhados ao módulo de mineração, responsável pela extração das regras de associação e pela apresentação das medidas de interesse, incluindo Suporte, Confiança e *Lift*. A análise temporal é realizada apenas para as regras previamente selecionadas pelo usuário na análise da base completa. Em cada partição temporal, a ferramenta avalia essas mesmas regras e apresenta suas medidas de interesse.

Por fim, o módulo de pós-processamento é responsável pela organização dos resultados gerados durante a mineração e pela construção das visualizações gráficas utilizadas na análise temporal.

Esse módulo permite comparar os valores das medidas obtidas nas diferentes partições temporais com os resultados encontrados na base completa, auxiliando na interpretação da evolução temporal dos padrões identificados. Quando uma regra não é encontrada em determinada partição, suas medidas são apresentadas com valor zero.

3.2 Fluxo de Execução

O fluxo de utilização da ferramenta se inicia com o carregamento de um arquivo CSV contendo os dados a serem analisados. Após a importação da base, a ferramenta apresenta automaticamente um resumo geral dos dados carregados, incluindo o número total de registros e a quantidade de atributos disponíveis na base de dados.

A ferramenta utiliza como entrada arquivos CSV contendo os dados extraídos dos repositórios de software a serem analisados. Considerando que o algoritmo *Apriori* opera sobre atributos categóricos, os atributos da base devem estar representados nesse formato para participarem da geração das regras de associação. Assim, atributos contínuos e atributos do tipo data devem ser previamente discretizados antes da importação dos dados. Durante o carregamento da base, a ferramenta identifica e destaca os diferentes tipos de atributos presentes no conjunto de dados, auxiliando o usuário na validação das informações utilizadas na análise.

Na sequência, a ferramenta apresenta um resumo dos possíveis valores de cada atributo, juntamente com suas respectivas frequências na base de dados. Essas frequências podem ser visualizadas tanto de forma geral quanto temporalmente por meio de gráficos, permitindo que o usuário compreenda o comportamento e a distribuição dos dados antes da extração das regras.

A Figura 2 apresenta um exemplo da visualização exploratória disponibilizada pela ferramenta para o atributo *first pull*. Na parte superior da figura é exibida a distribuição geral dos valores do atributo considerando toda a base de dados analisada. Já na parte inferior são apresentadas barras verticais que representam distribuições trimestrais (*quarterly*) mostrando a evolução da frequência de

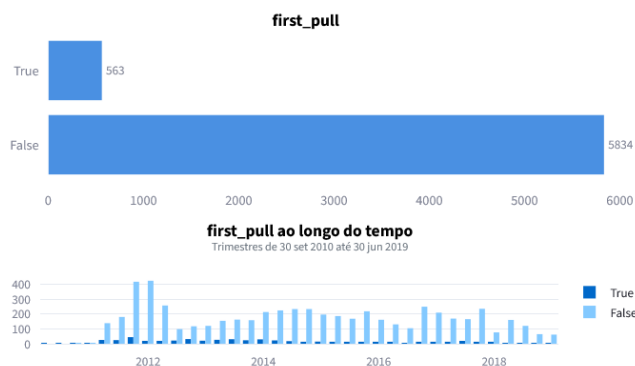


Figura 2: Visualização da frequência geral e evolução temporal do atributo *first pull* na ferramenta.

cada possível valor do atributo ao longo do tempo. Essa visualização auxilia na compreensão da distribuição dos dados antes da etapa de mineração das regras de associação.

A ferramenta também permite que o usuário defina, no início da análise, a ordem de exibição dos valores nos gráficos. Essa funcionalidade é importante em atributos cujos valores possuem uma relação semântica ou ordinal. Por exemplo, no atributo *lifetime*, os valores podem representar diferentes faixas de duração, como *very short*, *short*, *medium* e *lengthy*. Sem esse ajuste, os valores seriam apresentados na ordem em que aparecem originalmente na base de dados, o que pode dificultar a interpretação visual da evolução temporal dos padrões.

Após a etapa exploratória inicial, o usuário acessa a aba de geração de regras gerais, na qual são definidos os parâmetros mínimos de Suporte e Confiança utilizados na mineração das regras de associação da base completa. Nessa etapa, o usuário também pode selecionar quais regras deseja acompanhar na análise temporal, evitando o processamento desnecessário de grandes quantidades de regras e reduzindo o custo computacional da análise.

Em seguida, o usuário acessa a aba de análise temporal, responsável pela configuração das estratégias de particionamento da base de dados. Atualmente, a ferramenta oferece três estratégias de particionamento: i) definição manual de marcos temporais; ii) intervalos temporais de mesmo tamanho; iii) intervalos temporais com mesma quantidade de registros.

Após definir a estratégia de particionamento, o usuário pode reconfigurar os valores mínimos de Suporte e Confiança para a geração das regras nas partições temporais.

Finalizada essa configuração, a ferramenta executa a mineração das regras em cada partição e apresenta os resultados da análise temporal por meio de gráficos. Para cada regra selecionada, são exibidos três gráficos correspondentes às medidas de Suporte, Confiança e *Lift*. Esses gráficos apresentam simultaneamente os valores obtidos em cada partição temporal e os valores encontrados na base completa, permitindo comparar a estabilidade ou variação das regras ao longo do tempo.

Os gráficos da ferramenta são interativos e apresentam informações detalhadas por meio de *tooltips*. Ao posicionar o cursor

sobre uma barra, são exibidos o período correspondente à partição (datas de início e fim), a quantidade de registros da partição e os valores de Suporte, Confiança e *Lift* da regra analisada. Dessa forma, o tamanho e as características de cada partição podem ser consultados durante a exploração dos resultados, sem comprometer a legibilidade das visualizações.

Além das medidas das regras, a ferramenta também apresenta gráficos contendo a evolução temporal do suporte do antecedente e do consequente individualmente. Essa visualização complementar auxilia na interpretação das mudanças observadas nas medidas de interesse, permitindo identificar se determinadas variações estão relacionadas ao comportamento isolado do antecedente, do consequente ou da associação entre ambos.

3.3 Tecnologias Utilizadas

A ferramenta foi desenvolvida utilizando a linguagem Python devido à ampla disponibilidade de bibliotecas voltadas para mineração de dados, análise estatística e visualização de informações. A interface da aplicação foi implementada utilizando o framework Streamlit, permitindo a construção de uma interface interativa acessível via navegador e voltada à exploração visual dos dados e das regras de associação.

Para a mineração das regras de associação foi utilizada a biblioteca MLxtend, responsável pela implementação do algoritmo *Apriori* e pela geração das regras e medidas de interesse Suporte, Confiança e *Lift*.

A manipulação e o processamento dos dados são realizados principalmente com o auxílio da biblioteca Pandas⁵. Já as visualizações gráficas são construídas utilizando as bibliotecas Matplotlib⁶ e Plotly⁷, permitindo a geração de gráficos utilizados ao longo da análise temporal. O uso combinado dessas tecnologias possibilitou a construção de uma ferramenta interativa voltada à exploração temporal de regras de associação em dados de repositórios de software.

Foi utilizada a ferramenta de inteligência artificial generativa ChatGPT⁸ para revisão e aprimoramento textual do artigo.

3.4 Funcionalidades

A ferramenta oferece suporte à análise temporal de regras de associação em dados de repositórios de software, integrando recursos de mineração de dados, particionamento temporal e visualização interativa dos resultados em um único ambiente.

Entre suas principais funcionalidades estão:

- (1) visualização da distribuição temporal dos atributos da base;
- (2) seleção de regras específicas para análise temporal;
- (3) suporte a particionamento temporal;
- (4) mineração de regras de associação;
- (5) comparação das medidas obtidas na base completa e nas partições temporais;
- (6) visualização gráfica da evolução temporal das regras;
- (7) visualização complementar do comportamento temporal do antecedente e consequente das regras.

⁵<https://pandas.pydata.org>

⁶<https://matplotlib.org>

⁷<https://plotly.com>

⁸<https://chatgpt.com>

3.5 Usuário Potenciais

A ferramenta foi desenvolvida para apoiar pesquisadores, desenvolvedores e gerentes de projeto na análise de dados provenientes de repositórios de software. Seu público-alvo inclui pesquisadores das áreas de Mineração de Repositórios de Software e Engenharia de Software, estudantes de pós-graduação e profissionais interessados em compreender padrões relacionados ao desenvolvimento colaborativo de projetos de código aberto.

Nesse contexto, a proposta busca facilitar a exploração temporal de regras de associação extraídas de dados de projetos de software. Os recursos de particionamento temporal e visualização gráfica auxiliam análises relacionadas à evolução de padrões de contribuição, revisões, aceitação de *pull requests* e processos colaborativos, apoiando estudos empíricos com interpretação temporal dos padrões encontrados.

4 Cenário de Demonstração

Esta seção apresenta um cenário de demonstração da ferramenta, ilustrando sua utilização na análise temporal de regras de associação extraídas de dados de projetos de código aberto do GitHub. São mostrados exemplos de aplicação e visualizações que auxiliam na interpretação das variações temporais identificadas.

4.1 Aplicação em Projetos Reais

Para demonstrar o funcionamento da ferramenta, foram utilizados dados de *pull requests* provenientes de projetos de código aberto previamente analisados em um estudo sobre o impacto da criação e evolução do arquivo *contributing.md* em padrões de contribuição [6]. Os projetos selecionados possuem histórico temporal de modificações nesse arquivo, permitindo investigar possíveis mudanças nos padrões extraídos das regras de associação ao longo do tempo.

Os dados analisados incluem informações relacionadas aos *pull requests*, colaboradores e resultados das revisões realizadas nos projetos. A ferramenta foi utilizada para extrair regras de associação a partir da base completa e comparar temporalmente as medidas das regras em diferentes partições temporais, definidas a partir de marcos temporais correspondentes à criação e às modificações do arquivo *contributing.md*.

4.2 Exemplos de Variações Temporais

A Figura 3 apresenta um exemplo de análise temporal realizada pela ferramenta para a regra *FirstPull* = “true” → *Merged* = “false” no projeto Metasploit⁹. Essa regra indica uma associação entre *pull requests* que representam a primeira contribuição de um desenvolvedor e a rejeição dessas contribuições.

Para a realização da análise temporal apresentada na Figura 3, foi utilizado o particionamento baseado na definição de marcos temporais relacionados à inserção, em 06/11/2012, e à posterior alteração, em 25/02/2014, do arquivo *contributing.md* no projeto.

O gráfico ilustra a evolução temporal da medida *Lift* da regra nas partições resultantes do particionamento. Ao considerar apenas o valor obtido na análise geral, representado pela linha tracejada vermelha (*Lift*=2,36), parte importante do comportamento temporal da regra deixaria de ser observada.

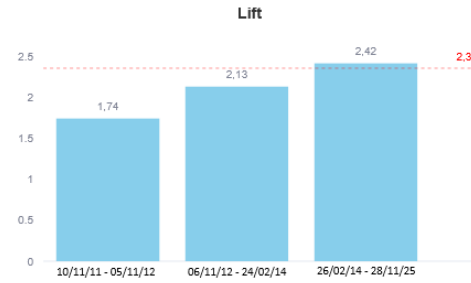


Figura 3: Análise temporal da medida de *Lift* da regra: *FirstPull* = “true” → *merged* = “false” no projeto Metasploit.

Na primeira partição temporal, anterior à inserção do arquivo *contributing.md*, a regra apresentou *Lift*=1,74, indicando que ser a primeira contribuição de um desenvolvedor aumentava em aproximadamente 74% a chance de rejeição do *pull request*. Após a inserção do arquivo, observa-se um aumento do *Lift* para 2,13, indicando fortalecimento da associação. Posteriormente, após a alteração representada pelo segundo marco temporal, o valor do *Lift* atingiu 2,42, ultrapassando o valor observado na análise geral da base.

A Figura 4 apresenta uma análise temporal da regra Tipo de Desenvolvedor = “comunidade externa” → Lifetime = “lengthy” extraída do projeto React¹⁰. A análise foi realizada utilizando particionamento baseado em marcos temporais relacionados à evolução do arquivo *contributing.md* no projeto.

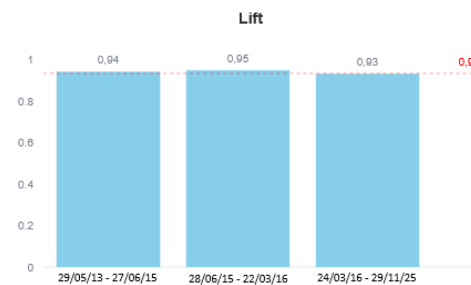


Figura 4: Análise temporal da medida de *Lift* da regra: Tipo de Desenvolvedor = “comunidade externa” → Lifetime = “lengthy” no projeto React.

Diferentemente do exemplo anterior, observa-se que os valores da medida de *Lift* permaneceram relativamente estáveis ao longo das partições temporais analisadas, variando entre 0,93 e 0,95, valores bastante próximos ao obtido na análise geral da base (*Lift*=0,94).

Como os valores de *Lift* permaneceram próximas a 1 em todas as partições, a regra analisada apresenta uma pequena associação negativa estável ao longo do tempo, indicando que o antecedente da regra reduz consistentemente a probabilidade de ocorrência do consequente no projeto analisado.

⁹<https://github.com/rapid7/metasploit-framework>

¹⁰<https://github.com/facebook/react>

Esse resultado demonstra que a ferramenta também permite identificar padrões temporalmente consistentes, auxiliando pesquisadores na distinção entre regras que sofrem alterações significativas ao longo do tempo e regras que permanecem estáveis mesmo após mudanças relevantes no projeto.

A Figura 5 apresenta a tela da ferramenta durante uma análise temporal da regra $\text{FirstPull} = \text{"true"} \rightarrow \text{Merged} = \text{"true"}$, que representa a associação entre ser a primeira contribuição e a aceitação do *pull requests*. À esquerda, a interface exibe o menu de navegação da análise e, na região central, os gráficos temporais das medidas de interesse e dos suportes do antecedente e consequente da regra.

A análise temporal utilizou particionamento baseado em marcos relacionados à inserção e evolução do arquivo *contributing.md*. Os gráficos superiores apresentam a evolução temporal das medidas de Suporte, Confiança e *Lift* da regra, além dos valores da análise geral da base, representados pelas linhas tracejadas vermelhas.

A ilustração apresenta crescimento gradual da medida de Confiança ao longo das partições temporais. Na primeira partição, a Confiança era de 0,51, indicando menor proporção de aceitação das primeiras contribuições, enquanto na última atingiu 0,95, evidenciando aumento expressivo na aceitação desse tipo de *pull request*.

Observa-se também um aumento gradual do valor de *Lift* ao longo das partições temporais analisadas. Na primeira partição, o valor de *Lift*=0,64 indicava uma associação negativa entre ser a primeira contribuição de um desenvolvedor e a aceitação do *pull request*. Após os marcos temporais definidos para o particionamento, o *Lift* aumentou progressivamente até atingir 0,99 na última partição, valor próximo da independência entre antecedente e consequente.

Além das medidas da regra, a ferramenta apresenta o comportamento temporal do antecedente e consequente na metade inferior da Figura 5. Nesse exemplo, observa-se redução do suporte de $\text{FirstPull} = \text{"true"}$ e aumento do suporte de $\text{Merged} = \text{"true"}$ ao longo do

tempo. Essas visualizações auxiliam na interpretação das possíveis causas das variações observadas nas medidas da regra.

5 Conclusões e Trabalhos Futuros

Este artigo apresenta uma ferramenta para análise temporal de regras de associação em dados de projetos de software. A abordagem permite extrair regras utilizando o algoritmo *Apriori*, aplicar particionamento temporal e visualizar a evolução das medidas de interesse por meio de gráficos.

Os exemplos apresentados com dados reais de projetos de código aberto demonstraram o potencial da ferramenta para auxiliar pesquisadores, desenvolvedores e gerentes de projetos de software na identificação e interpretação de mudanças temporais em padrões relacionados a *pull requests* e processos colaborativos.

Como limitação, a ferramenta não identifica automaticamente os períodos mais relevantes para análise nem aponta as causas das variações observadas nas medidas das regras, cabendo ao usuário definir os particionamentos temporais e analisar os resultados.

Em trabalhos futuros pretende-se adicionar novos mecanismos de particionamento temporal e funcionalidades automáticas para identificação de variações significativas nas medidas das regras, além de realizar estudos experimentais com usuários para avaliar a usabilidade e utilidade da ferramenta.

Disponibilidade de Artefatos

Os artefatos gerados por esta pesquisa estão disponíveis publicamente em <https://doi.org/10.5281/zenodo.22212565>. O código-fonte do TEMAR, licenciado sob a licença MIT, está disponível no GitHub em <https://github.com/silvana21/toolTemp>, cujo repositório inclui instruções de instalação, dependências, guia de utilização e bases de dados de exemplo. A demonstração em vídeo está disponível em <https://doi.org/10.5281/zenodo.20367413>.



Figura 5: Tela da ferramenta de análise temporal da regra: $\text{FirstPull} = \text{"true"} \rightarrow \text{Merged} = \text{"true"}$ no projeto Puppet.

Referências

- [1] Rakesh Agrawal and Ramakrishnan Srikant. 1994. Fast Algorithms for Mining Association Rules in Large Databases. In *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB '94)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 487–499.
- [2] Michael R. Berthold, Nicolas Cebron, Fabian Dill, Thomas R. Gabriel, Tobias Kötter, Thorsten Meinl, Peter Ohl, Kilian Thiel, and Bernd Wiswedel. 2009. KNIME - the Konstanz information miner: version 2.0 and beyond. *SIGKDD Explor. Newsl.* 11, 1 (Nov. 2009), 26–31. doi:10.1145/1656274.1656280
- [3] Juan Andrés Carruthers, Jorge Andrés Diaz-Pace, and Emanuel Irrazábal. 2024. A longitudinal study on the temporal validity of software samples. *Information and Software Technology* 168 (2024), 107404. doi:10.1016/j.infsof.2024.107404
- [4] Janez Demšar, Tomaž Curk, Aleš Erjavec, Črt Gorup, Tomaž Hočevar, Mitar Milutinović, Martin Možina, Matija Polajnar, Marko Toplak, Anže Starič, et al. 2013. Orange: data mining toolbox in Python. *the Journal of machine Learning research* 14, 1 (2013), 2349–2353.
- [5] Silvana Gonçalves, Daricélio Soares, and Daniel Silva. 2021. Temporal analysis on pull request patterns: an approach with sliding window. In *Proceedings of the 15th Brazilian Symposium on Software Components, Architectures, and Reuse (Joinville, Brazil) (SBCARS '21)*. Association for Computing Machinery, New York, NY, USA, 90–99. doi:10.1145/3483899.3483906
- [6] Silvana Gonçalves, Alexandre Plastino, and Daricélio Soares. 2025. Avaliando a Importância do Arquivo contributing.md em Projetos de Código Aberto. In *Anais do LII Seminário Integrado de Software e Hardware (Maceió/AL)*. SBC, Porto Alegre, RS, Brasil, 13–24. doi:10.5753/semish.2025.6870
- [7] Geoffrey Holmes, Andrew Donkin, and Ian H Witten. 1994. Weka: A machine learning workbench. In *Proceedings of ANZIS'94-Australian New Zealand Intelligent Information Systems Conference*. IEEE, Institute of Electrical and Electronics Engineers, Brisbane, QLD, Australia, 357–361.
- [8] Ross Ihaka and Robert Gentleman. 1996. R: A Language for Data Analysis and Graphics. *Journal of Computational and Graphical Statistics* 5, 3 (1996), 299–314. arXiv:https://www.tandfonline.com/doi/pdf/10.1080/10618600.1996.10474713 doi:10.1080/10618600.1996.10474713
- [9] Daricélio Moreira Soares, Manoel Limeira de Lima Júnior, Leonardo Murta, and Alexandre Plastino. 2021. What factors influence the lifetime of pull requests? *Software: Practice and Experience* 51, 6 (2021), 1173–1193.
- [10] Daricélio Moreira Soares, Manoel Limeira de Lima Júnior, Leonardo Murta, and Alexandre Plastino. 2015. Acceptance factors of pull requests in open-source projects. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing (Salamanca, Spain) (SAC '15)*. Association for Computing Machinery, New York, NY, USA, 1541–1546. doi:10.1145/2695664.2695856
- [11] Daricélio Moreira Soares, Manoel L. De Lima Junior, Leonardo Murta, and Alexandre Plastino. 2015. Rejection Factors of Pull Requests Filed by Core Team Developers in Software Projects with High Acceptance Rates. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*. Institute of Electrical and Electronics Engineers, Miami, FL, USA, 960–965. doi:10.1109/ICMLA.2015.41
- [12] Daricélio M Soares, Manoel L de Lima Júnior, Alexandre Plastino, and Leonardo Murta. 2018. What factors influence the reviewer assignment to pull requests? *Information and Software Technology* 98 (2018), 32–43.
- [13] Mairieli Wessel, Joseph Vargovich, Marco A Gerosa, and Christoph Treude. 2023. Github actions: the impact on the pull request process. *Empirical Software Engineering* 28, 6 (2023), 131.