INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PIAUÍ
CAMPUS FLORIANO
CURSO TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

AUGUSTO FRANCO FERREIRA SOARES

UMA REVISÃO SISTEMÁTICA DA LITERATURA SOBRE COMITÊS DE CLASSIFICADORES PARA PREDIÇÃO DA ESTIMATIVA DO ESFORÇO DE EQUIPE DE *SOFTWARE*

FLORIANO-PI

2023

AUGUSTO FRANCO FERREIRA SOARES

UMA REVISÃO SISTEMÁTICA DA LITERATURA SOBRE COMITÊS DE CLASSIFICADORES PARA PREDIÇÃO DA ESTIMATIVA DO ESFORÇO DE EQUIPE DE *SOFTWARE*

Trabalho de Conclusão de Curso (artigo científico) apresentado como exigência parcial para obtenção do diploma do Curso de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Piauí, Campus Floriano.

Orientador: Prof. Msc. Francisco Eduardo Pires de Morais
Coorientador: Prof. Msc. Wilamis Kleiton Nunes da Silva

FLORIANO

2023

UMA REVISÃO SISTEMÁTICA DA LITERATURA SOBRE COMITÊS DE CLASSIFICADORES PARA PREDIÇÃO DA ESTIMATIVA DO ESFORÇO DE EQUIPE DE *SOFTWARE*

Augusto Franco Ferreira Soares[1]
Francisco Eduardo Pires de Morais[2]
Wilamis Kleiton Nunes da Silva[3]

RESUMO

A Predição da estimativa de esforço de equipe de *software* é uma das principais partes do ciclo de vida de um projeto. A predição desta estimativa ainda é uma etapa desafiadora e várias técnicas têm sido utilizadas ao longo dos anos, dentre elas destaca-se o uso de comitês de classificadores como uma técnica promissora. Portanto, o objetivo do estudo foi realizar uma revisão sistemática da literatura (RLS) sobre o uso de comitês de classificadores para predição da estimativa do esforço de equipe de *software*. Para a condução do processo da RSL utilizou-se como metodologia o estudo de Kitchenham (2009), as fontes de buscas utilizadas para a coleta dos estudos foram: IEEE Xplore, Scopus, ACM, e *Web of Science*, cujos trabalhos foram publicados nos anos de 2019 a 2023. Nos estudos coletados observa-se o uso predominantemente do dataset *Desharnais*, dentre as ferramentas/*frameworks* destacou-se o Weka como o mais empregado, entre as métricas de avaliação para predição da estimativa de esforço de *softwares* a mais aplicada foi PRED, a técnica de validação mais abordada foi o *Croos Validation*. Na avaliação da qualidade dos estudos, o estudo de Rai, Kumar e Verma (2021) obteve a maior soma de pontuação. Cada técnica de aprendizado de máquina utilizada nos estudos possui seus pontos positivos e negativos, portanto não foi possível afirmar qual é a melhor técnica.

Palavras-chave: Conjunto Heterogêneo; Conjunto Homogêneo; Esforço de Desenvolvimento de Software; Aprendizado de Máquina; Inteligência Artificial.

ABSTRACT

The Prediction of *software* team effort estimation is one of the key aspects of a project's life cycle. Predicting this estimate remains a challenging step, and various techniques have been used over the years, with the use of classifier committees standing out as a promising technique. Therefore, the aim of this study was to conduct a systematic literature review (SLR) on the use of classifier committees for predicting *software* team effort estimation. The Kitchenham (2009) study methodology was used to conduct the SLR process, and the search sources used to gather the studies were IEEE Xplore, Scopus, ACM, and Web of Science, with the studies published from 2019 to 2023. In the collected studies, the Desharnais dataset was predominantly used, and the Weka tool/framework was the most commonly employed.

[1] Graduando do curso de Tecnologia em Análise e Desenvolvimento de Sistemas do IFPI Floriano. caflo2021114tads05@aluno.ifpi.edu.br.
[2] Mestre em Engenharia de Software CESAR School. IFPI-Campus Floriano. professoreduardo.pires@ifpi.edu.br.
[3] Mestre em Ciência da Computação – UFERSA . Colégio Técnico de Floriano. wilamiskleiton@ufpi.edu.br.

The most applied evaluation metric for *software* effort estimation prediction was PRED, and the most addressed validation technique was Cross Validation. In the assessment of study quality, the study by Rai, Kumar, and Verma (2021) obtained the highest total score. Each machine learning technique used in the studies has its strengths and weaknesses, so it was not possible to determine the best technique.

Keywords: Heterogeneous Ensemble; Homogeneous Ensemble; Software Development Effort; Machine Learning; Artificial Intelligence

Data de aprovação: 20/11/2023 (data de apresentação do TCC)

1 INTRODUÇÃO

A crescente complexidade e escala dos projetos de desenvolvimento de *software* têm levado à necessidade de abordagens eficazes para gerenciar o esforço de equipe de *software* (RHMANN; PANDEY e ANSARI, 2021). O esforço de equipe de *software*, por sua vez, é um fator crítico para o sucesso de um projeto de *software*, uma vez que envolve a colaboração de diversos membros da equipe para atingir um objetivo comum (HAI, 2022).

Uma abordagem promissora que vem se destacando ao longo dos anos para melhorar a colaboração e a eficiência no desenvolvimento de *software* é o uso de comitês de classificadores. Os comitês de classificadores segundo a literatura são uma técnica de aprendizado de máquina que combina as predições de vários modelos individuais para obter uma predição mais precisa. O uso de comitê de classificadores vem cada vez mais ganhando seu espaço frente a técnicas tradicionais de aprendizado de máquinas (ARAÚJO, 2019).

Estudos foram realizados fazendo uso de comitês de classificadores na estimativa de esforço de equipe de *software* (GOYAL ,2022; CABRAL e OLIVEIRA,2021). Nessas pesquisas foram feitas comparações entre modelos de técnicas propostas, onde os mesmos são avaliados por métricas de avaliação usando determinados *datasets* para medir a precisão dessas técnicas.

É notório ainda que, algumas pesquisas deixam a desejar quando o assunto é fornecer detalhes para que novos trabalhos possam surgir dando continuidade a essas pesquisas (NOSRATI e RAHMANI, 2023; SAKHRAWI; SELLAMI e BOUASSIDA, 2022; KUMAR e YERESIME, 2023). Para que estudos com essa temática continuem sendo desenvolvidos, é necessário que os pesquisadores detalhem ainda mais os seus estudos haja vista a importância do desenvolvimento de estudos voltados para essa área.

Neste contexto, o objetivo deste trabalho foi realizar uma revisão sistemática da literatura (RLS) sobre o uso de comitês de classificadores para predição da estimativa do esforço de equipe de *software*, com o intuito de identificar possíveis lacunas para pesquisas futuras. Desta forma, a RSL poderá contribuir na área estudada fornecendo *insights* valiosos sobre as práticas, as abordagens e os resultados obtidos. Este estudo está organizado da seguinte forma: a Seção 2 explana sobre a metodologia do protocolo, a Seção 3 apresenta o referencial teórico, a Seção 4 aborda resultados do estudo, e, por fim, a Seção 5 traz as considerações finais do estudo.

2 REFERENCIAL TEÓRICO

Nesta seção, serão explorados os conceitos e fundamentos teóricos fundamentais relacionados aos subtemas "Esforço de Equipe de *Software*" e "Comitês de Classificadores", fornecendo uma base sólida para a compreensão desses elementos no contexto do estudo.

2.1 ESFORÇO DE EQUIPE DE *SOFTWARE*

No âmbito do desenvolvimento de *software*, a estimativa de esforço da equipe desempenha um papel crucial na fase inicial do projeto. Identificar com precisão a quantidade de trabalho necessária não apenas contribui para um planejamento mais realista, mas também impacta diretamente na alocação de recursos e no cronograma do projeto (JADHAV; KAUR e AKTER, 2022).

O esforço de equipe de *software*, no contexto do desenvolvimento de projetos, é uma dimensão essencial que abrange fases relevantes para planejamento e construção de um software (MAHMOOD, 2022). Esta colaboração vai além da simples distribuição de tarefas e envolve aspectos como coordenação, comunicação e compartilhamento de conhecimento. Com o crescente aumento na robustez e escala dos projetos de *software* vem a importância de uma abordagem eficaz para gerenciar o esforço de equipe, pois influencia diretamente no sucesso do projeto (SREEKANTH, 2023).

O esforço de equipe no desenvolvimento de *software* não é meramente uma questão quantitativa, mas sim uma interação qualitativa entre os membros da equipe de desenvolvimento, exercendo um impacto direto na qualidade e eficiência do processo de desenvolvimento do software (JADHAV, 2023). Dessa forma, predizer de forma concisa esse esforço realizado pela equipe de desenvolvimento é fundamental para as fases iniciais de planejamento de um produto de software.

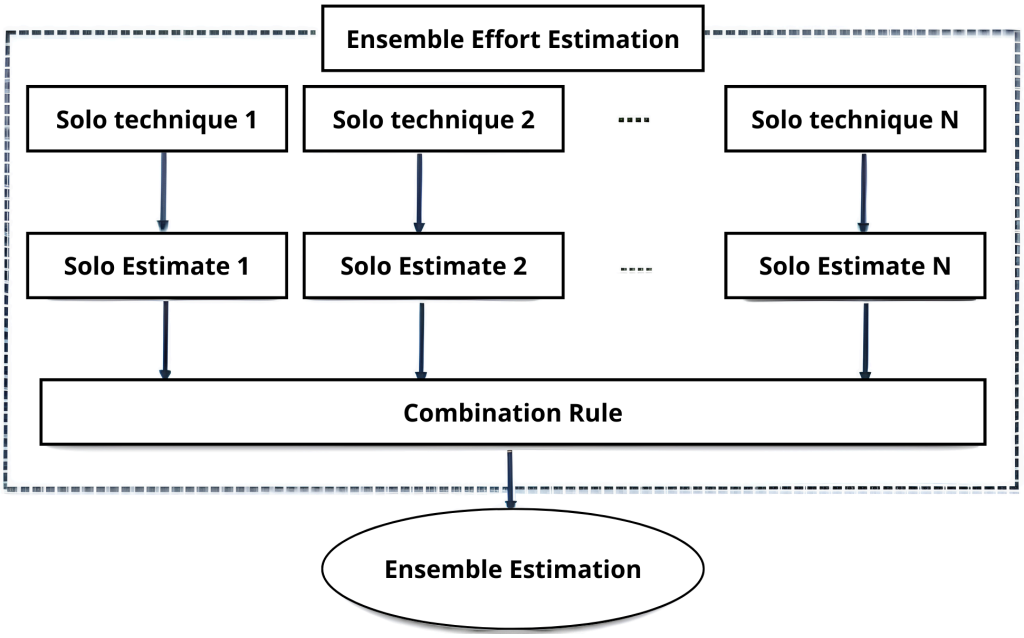
A princípio, os estudos acerca dessa temática que se propuseram a desenvolver métodos para se estimar o esforço realizados pelas equipes de software se baseiam em estimativa por analogia e também por técnicas de aprendizado de máquinas individuais (KUMAR, 2022). Como a temática já é discutida a muito tempo, novas abordagens surgiram no decorrer das últimas décadas o que levanta a possibilidade de técnicas mais eficientes serem utilizadas no contexto contemporâneo e que contribuam significativamente para a estimativa de esforço de equipe de software.

Sob essa perspectiva, fica notória a necessidade de investigar quais técnicas a literatura tem utilizado para construção de estimadores de esforço de *software*. Além disso, é crucial destacar que a compreensão aprofundada do esforço de equipe é fundamental para a implementação de práticas de gestão eficazes. Transitar da teoria para a prática envolve não apenas reconhecer a importância do esforço de equipe, mas também explorar as nuances e desafios enfrentados pelos desenvolvedores no contexto específico de cada projeto.

2.2 COMITÊS DE CLASSIFICADORES

No que diz respeito aos comitês de classificadores, essa abordagem representa uma evolução significativa no campo do aprendizado de máquina, especialmente quando aplicada à predição da estimativa do esforço de equipe de *software* (RAI; KUMAR e VERMA ,2021). Os comitês de classificadores são fundamentados na ideia de combinar as previsões de vários modelos individuais para obter uma estimativa mais precisa e confiável.

Figura 1: Esquema de funcionamento de um comitê de classificadores para estimativa de esforço de equipe de *software*.



Fonte: Idri, Hosni e Abran. (2016).

Observando a Figura 1 é possível visualizar de forma ilustrativa como funciona um comitê de classificadores para estimativa de esforço de *software*, onde às técnicas de aprendizado de máquina individuais são combinadas com outras técnicas de aprendizado de máquina e submetidas a uma regra de combinação para que os dados de saída de cada uma das técnicas sejam combinados e assim retornar uma estimativa mais eficaz (IDRI; HOSNI e ABRAN, 2016).

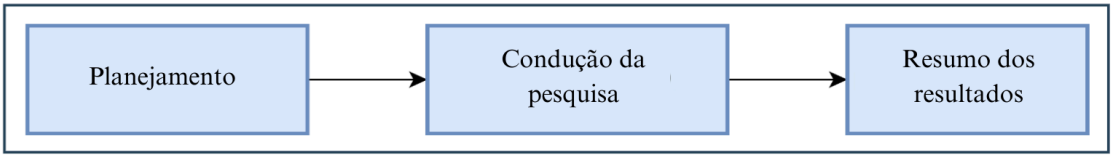
No contexto deste estudo, os comitês de classificadores são vistos como uma estratégia poderosa para lidar com a complexidade inerente à estimativa do esforço de equipe. Ao integrar diferentes perspectivas modelares, essa abordagem busca mitigar a incerteza e a variabilidade presentes nos ambientes de desenvolvimento de *software*. A literatura destaca que o uso de comitês de classificadores tem se consolidado como uma alternativa viável e eficaz em comparação com as abordagens tradicionais de aprendizado de máquina (SHUKLA e KUMAR, 2023; MARAPELLI; CARIE e ISLAM ,2021; BEESETTI; BILGAIYAN e MISHRA, 2023).

Ao estabelecer uma base teórica sólida para os subtemas "Esforço de Equipe de *Software*" e "Comitês de Classificadores", esta seção proporciona os alicerces necessários para a análise crítica e interpretação dos resultados obtidos na Revisão Sistemática da Literatura (RSL). Essa análise crítica é essencial para a compreensão profunda das implicações práticas e teóricas desses temas no contexto do desenvolvimento de *software*.

3 METODOLOGIA

Para a condução do processo da RSL utilizou-se como metodologia o estudo de Kitchenham (2009) uma referência na área da ciência da computação no desenvolvimento de RSL. Para alcançar os objetivos do estudo foram executadas as seguintes etapas: Planejamento da Pesquisa, Condução da Pesquisa e Síntese dos Resultados, conforme Figura 2.

Figura 2: Etapas da RSL



Fonte: Adaptado de Kitchenham (2009)

Na definição do planejamento da pesquisa, foram definidas as questões de pesquisa, além da definição do protocolo. Na fase da condução da pesquisa realizou-se a seleção dos estudos aplicando a *string* de busca nas bibliotecas digitais selecionadas seguindo os critérios da filtragem dos estudos com base nos critérios de inclusão, extração e sintetização dos dados dos estudos selecionados.

3.1 QUESTÕES DE PESQUISA

O objetivo do trabalho foi realizar uma revisão sistemática da literatura sobre o uso de comitês de classificadores para predição da estimativa do esforço de equipe de *software*. Para tanto, para alcançar o objetivo, foram construídas as seguintes questões de pesquisa:

Q1)	***Qual o datasets mais empregado nos estudos?***
Q2)	***Quais as ferramentas/framework mais utilizadas nos estudos?***
Q3)	***Qual a métrica de avaliação mais aplicada nos estudos?***
Q4)	***Qual a técnica de validação mais abordada nos estudos?***
Q5)	***Qual a melhor técnica de aprendizado de máquina empregada nos estudos?***

3.2 BASE DE DADOS

As fontes de buscas utilizadas para realização deste estudo foram escolhidas pelo grau de relevância científica considerando a área de concentração de ciências da computação e informática, por serem fontes reconhecidas pela comunidade científica com excelentes níveis de qualificação, bem como a acessibilidade dos trabalhos indexados. O Quadro 1 apresenta as bases de dados utilizadas para a coleta dos estudos.

Quadro 1 - Bases de Dados

Fontes de Busca	URLs
IEEE Xplore	http://www.ieeexplore.ieee.org/Xplore
Scopus	http://www.scopus.com/
ACM	https://dl.acm.org/
Web of Science	https://www.webofscience.com/

Fonte: Elaborado pelos autores.

3.3 ESTRATEGIAS DE BUSCA

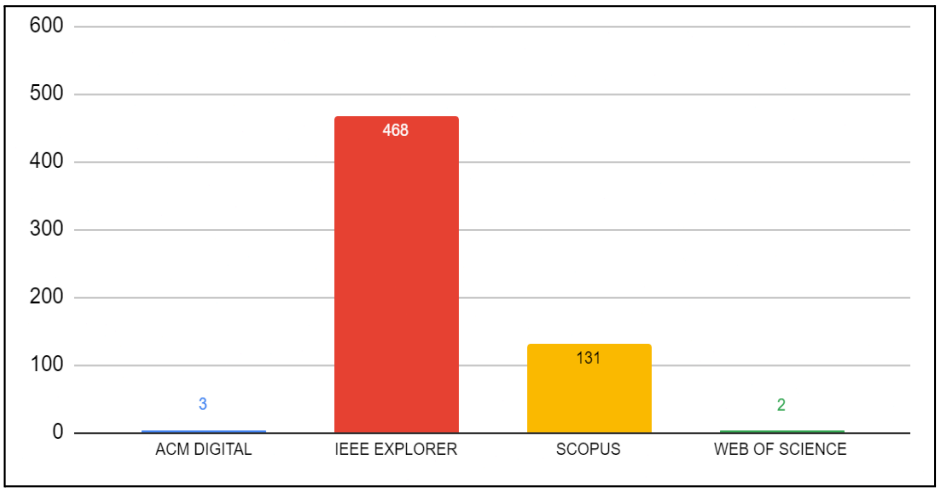
Para a coleta dos estudos, optou-se pela busca nas bibliotecas digitais, cujos trabalhos foram publicados nos anos de 2019 a 2023, em inglês, a escolha do idioma em inglês justifica-se devido à universalidade do idioma, a não inclusão do português se dá pela

necessidade de o estudo ser passível de reprodução em diferentes países. Para a RSL foi definida a *string* de busca:

> *(“Heterogeneous Ensemble” **OR** “Homogeneous Ensemble”) **AND** ("Software Development Effort" **OR** "Software Project Effort") **AND** ("Machine Learning" **OR** "Artificial Intelligence")*

Aplicando a busca nas bases de dados através *string* de busca foram retornados um total de 604 estudos, apresentados na Figura 3, sendo possível observar, que a biblioteca digital IEEE *Xplore* apresentou a maior quantidade de estudos depositados, enquanto a biblioteca digital *Web of Science* contém a menor quantidade de estudos.

Figura 3: Bibliotecas Digitais Utilizadas para Coleção de Estudos



Fonte: Elaborada pelos autores.

Depois da aplicação dos critérios de exclusão e leitura dos títulos, dos resumos e eventualmente do corpo do estudo, dos 604 estudos retornados, apenas 51 deles foram incluídos para a fase de leitura integral. A leitura dos estudos foi realizada em dupla a fim de minimizar qualquer possibilidade de análises tendenciosas, em caso de empate um terceiro avaliador foi convocado, após a conclusão da fase restaram 23 estudos para a fase de extração e síntese dos resultados, sendo 3 estudos da biblioteca digital *IEEE Xplore*, 18 estudos da *Scopus* e 2 estudos da *Web of Science.*

3.4 CRITÉRIOS DE INCLUSÃO E EXCLUSÃO

Com o intuito de responder às questões de pesquisa foram incluídos e excluídos estudos considerados relevantes e irrelevantes para a revisão sistemática. Os critérios de inclusão e exclusão foram:

3.4.1 Critérios de inclusão:

- Estudos completos publicados nos anos de 2019 a 2023;
- Estudos primários completos publicados em conferências, revistas e workshop que foram revisados por pares;
- Estudos que relatam o uso de comitê de classificadores para predição da estimativa do esforço de equipe de *software*;
- Trabalhos escritos em Inglês.

3.4.2 Critérios de exclusão:

- Trabalhos duplicados;
- Estudos não disponíveis para acesso online;
- Trabalhos publicados como *short-papers* e/ou estudos secundários, como outras revisões sistemáticas, surveys e capítulos de livros.

3.5 ANÁLISE DE QUALIDADE

A avaliação da qualidade dos estudos fornece critérios de inclusão/exclusão mais apurados. Todos os estudos selecionados foram analisados por completo. Os estudos foram confrontados com 08 questões e cada estudo recebeu uma nota de relevância com base na sua conformidade com as questões. O Quadro 2 apresenta as questões de análise da qualidade dos estudos.

Quadro 2 - Questões de análise da qualidade dos estudos

Numeração	Pergunta
Q1)	*O estudo relata o uso de comitê de classificador para predição da estimativa do esforço de equipe de software?*
Q2)	*Os objetivos do estudo estão claramente definidos?*
Q3)	*O contexto do estudo está adequadamente descrito?*
Q4)	*O projeto de pesquisa foi adequado para atingir os objetivos da pesquisa?*
Q5)	*Os resultados da pesquisa foram validados de forma adequada?*
Q6)	*O estudo contribui para a predição da estimativa de esforço de software?*
Q7)	*O estudo aponta quais as variáveis que geram maior impactos na predição da estimativa de esforço de software?*
Q8)	*O estudo apresenta teste estatístico?*

Fonte: Elaborado pelos autores.

4 RESULTADOS

Ao examinar de perto os estudos incluídos, observamos padrões consistentes relacionados às técnicas utilizadas para a estimativa de esforço de equipe de *software*, destacando as abordagens mais prevalentes e as metodologias mais frequentemente empregadas. Além disso, identificamos áreas em que as lacunas se manifestam, fornecendo uma base sólida para futuras investigações. A seguir é possível visualizar os resultados obtidos por meio da RSL.

A. QUAL O DATASETS MAIS EMPREGADO NOS ESTUDOS?

Datasets são componentes essenciais para o estudo de técnicas para predição da estimativa de esforço de equipe de *software*, pois permitem que as empresas tomem decisões mais eficientes, baseadas em fatos reais. A Tabela 1 apresenta os *datasets* empregados nos estudos.

Tabela 1: *Datasets* usados nos estudos

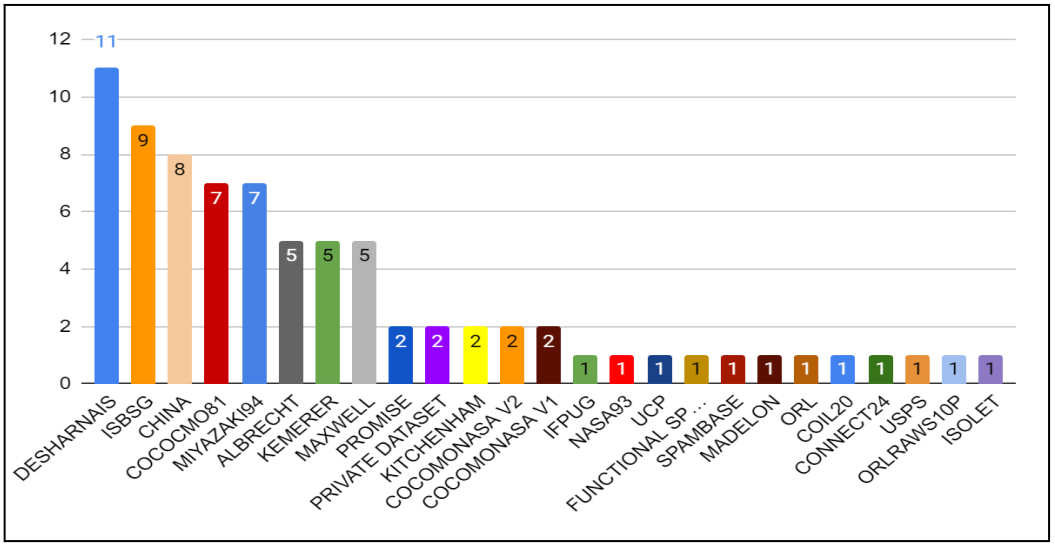
Dataset	**Estudos**	**Total**
DESHARNAIS	Marco, Ahmad e Ahmad (2022); Goyal (2022); Cabral e Oliveira (2021); Abnane et al. (2023); Hosni, Idri e Abran (2021); Beesetti, Bilgaiyan e Mishra (2023); Rahman, Pandey e Ansari (2021); Abnane et al. (2021); Shahpar, Bardsiri e Bardsiri (2022); Hosni, Idri e Abran (2019); Kumar e Yeresime (2023)	11
ISBSG	Marco, Ahmad e Ahmad (2022); Ali et al. (2023); Hosni, Idri e Abran (2021); Labidi e Sakhrawi (2023); Rai, Kumar e Verma (2021); Shukla e Kumar (2023); Sakhrawi, Sellami e Bouassida (2022); Abnane et al. (2021); Hosni, Idri e Abran (2019)	9
CHINA	Marco, Ahmad e Ahmad (2022); Goyal (2022); Cabral e Oliveira (2021); Abnane et al. (2023);Hosni, Idri e Abran (2021); Beesetti, Bilgaiyan e Mishra (2023); Abnane et al. (2021); Hosni, Idri e Abran (2019)	8
COCOCMO81	Marco, Ahmad e Ahmad (2022); Goyal (2022); Cabral e Oliveira (2021); Abnane et al. (2023); Hosni, Idri e Abran (2021); Beesetti, Bilgaiyan e Mishra (2023); Abnane et al. (2021);	7
MIYAZAKI94	Goyal (2022); Cabral e Oliveira (2021); Abnane et al. (2023); Hosni, Idri e Abran (2021); Abnane et al. (2021); Hosni, Idri e Abran (2019)	6
ALBRECHT	Marco, Ahmad e Ahmad (2022); Hosni, Idri e Abran (2021); Beesetti, Bilgaiyan e Mishra (2023); Shahpar, Bardsiri e Bardsiri (2022); Hosni, Idri e Abran (2019)	5
KEMERER	Marco, Ahmad e Ahmad (2022); Abnane et al. (2023); Beesetti, Bilgaiyan e Mishra (2023);, Abnane et al. (2021); Shahpar, Bardsiri e Bardsiri (2022)	5
MAXWELL	Marco, Ahmad e Ahmad (2022); Goyal (2022); Cabral e Oliveira (2021); Beesetti, Bilgaiyan e Mishra (2023); Shahpar, Bardsiri e Bardsiri (2022)	5
PROMISE	Shukla e Kumar (2023); Charmanas, Mittas e Angelis(2020)	2
PRIVATE DATASET	Marapelli, Carie e Islam (2021); Shukla e Kumar (2023)	2
KITCHENHAM	Marco, Ahmad e Ahmad (2022); Cabral e Oliveira (2021);	2
COCOMONASA V2	Cabral e Oliveira (2021); Rahman, Pandey e Ansari (2021);	2
COCOMONASA V1	Cabral e Oliveira (2021); Rahman, Pandey e Ansari (2021);	2
IFPUG	Marco, Ahmad e Ahmad (2022);	1
NASA93	Marco, Ahmad e Ahmad (2022);	1
UCP	Marco, Ahmad e Ahmad (2022);	1
FUNCTIONAL SPECIFICATION	Sakhrawi, Sellami e Bouassida (2022)	1
SPAMBASE	Nosrati e Rahmani (2023)	1

MADELON	Nosrati e Rahmani (2023)	1
ORL	Nosrati e Rahmani (2023)	1
COIL20	Nosrati e Rahmani (2023)	1
CONNECT24	Nosrati e Rahmani (2023)	1
USPS	Nosrati e Rahmani (2023)	1
ORLRAWS10P	Nosrati e Rahmani (2023)	1
ISOLET	Nosrati e Rahmani (2023)	1

Fonte: Elaborado pelos autores.

É possível observar na Tabela 1 que, os *datasets* predominantemente utilizadas nos estudos foram *Desharnais* com 47.83% dos casos (11/23), ISBSG com 39.13% dos casos (9/23), China com 34.78% dos casos (8/23), Cococmo81 com 30.43% dos casos (7/23) e Miyazaki94 com 26.09% dos casos (6/23), a escolha se justifica pelo fato dos *datasets* fornecerem estimativas generalizadas e melhores em relação aos demais *datasets*. A Figura 3 apresenta a distribuição dos *datasets* utilizados nos estudos.

Figura 4: Distribuição dos *datasets* utilizados no estudo



Fonte: Elaborado pelos autores.

B. QUAL A FERRAMENTA/FRAMEWORK MAIS UTILIZADA NOS ESTUDOS?

A Tabela 2 expõe as ferramentas/*framework* utilizadas nos estudos para a predição da estimativa de esforço de equipe de *software*. A ferramenta *Weka* foi utilizada em 2 estudos, o *Google Colab* foi utilizado em 2 estudos, a ferramenta RKEEL and *MetaheuristicsOpt R Packages* foi utilizada em 1 estudo, e por fim, o *Matlab* foi utilizado em 1 estudo. Ademais, vale ressaltar que, 18 estudos não relataram o uso de ferramentas/*framework* nos seus projetos.

Tabela 2: ferramenta/*framework* mais utilizado nos estudos

Ferramenta/*framework*	Estudos	Total
WEKA	Cabral e Oliveira (2021); Hosni, Idri e Abran (2021)	2
GOOGLE COLAB	Sakhrawi, Sellami e Bouassida (2022); Sakhrawi, Sellami e Bouassida (2022)	2
RKEEL AND METAHEURISTICSOPT R PACKAGES	Rahman, Pandey e Ansari (2021)	1
MATLAB	Goyal (2022)	1

Fonte: Elaborada pelos autores

C. QUAL A MÉTRICA DE AVALIAÇÃO MAIS APLICADA NOS ESTUDOS?

A Tabela 3 descreve as métricas de avaliação utilizadas nos estudos. As métricas buscam definir um método padrão para medir atributos do processo fornecendo o controle eficiente de um projeto de desenvolvimento de *software*.

Tabela 3: Métricas de avaliação utilizadas no estudo

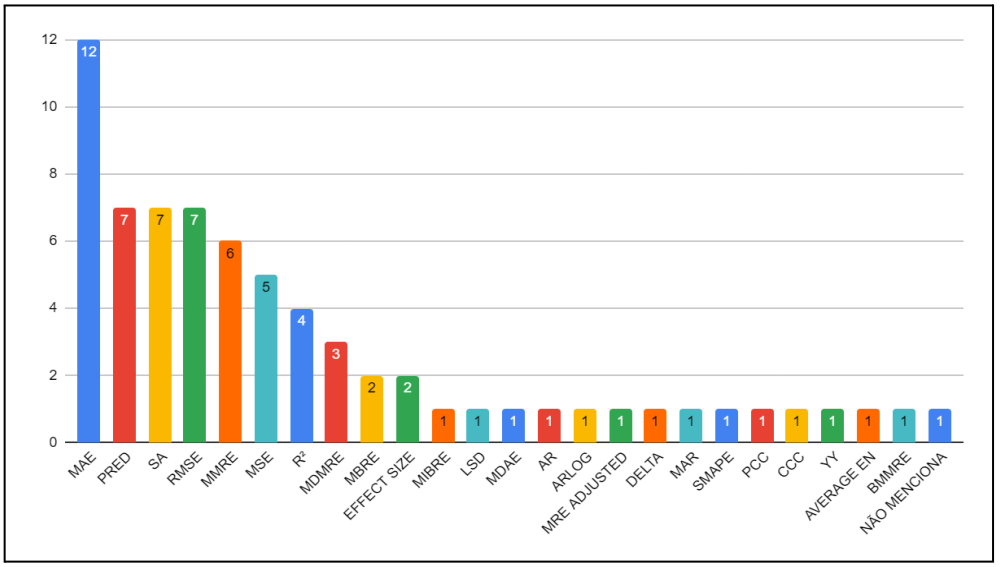
Métrica de avaliação	Menções	Total
MAE	Marco, Ahmad e Ahmad (2022);Goyal (2022); Ali et al. (2023); Labidi e Sakhrawi (2023); Rai, Kumar e Verma (2021); Shukla e Kumar (2023); Beesetti, Bilgaiyan e Mishra (2023); Rahman, Pandey e Ansari (2021); Sakhrawi, Sellami e Bouassida(2022); Sakhrawi, Sellami e Bouassida(2022); Shukla e Kumar (2023); Hosni, Idri e Abran (2019)	12
PRED	Goyal (2022); Abnane et al. (2023); Ali et al. (2023); Mahmood et al. (2020); Shukla e Kumar (2023); Shahpar, Bardsiri e Bardsiri (2022); Kumar e Yeresime (2023)	7
SA	Abnane et al. (2023); Hosni, Idri e Abran (2021);Shukla e Kumar (2023); Sakhrawi, Sellami e Bouassida(2022); Shukla e Kumar (2023); Abnane et al. (2021); Hosni, Idri e Abran (2019)	7
RMSE	Marco, Ahmad e Ahmad (2022); Labidi e Sakhrawi (2023); Rai, Kumar e Verma (2021); Beesetti, Bilgaiyan e Mishra (2023); Rahman, Pandey e Ansari (2021); Sakhrawi, Sellami e Bouassida(2022); Sakhrawi, Sellami e Bouassida(2022)	7
MMRE	Goyal (2022); Ali et al. (2023); Mahmood et al. (2020); Shahpar, Bardsiri e Bardsiri (2022); Srivastava, Sharma e Choudhary (2021); Kumar e Yeresime (2023)	6
MSE	Rai, Kumar e Verma (2021); Beesetti, Bilgaiyan e Mishra (2023); Marapelli, Carie e Islam (2021); Shukla e Kumar (2023); Hosni, Idri e Abran (2019)	5
R^2	Marco, Ahmad e Ahmad (2022); Labidi e Sakhrawi (2023); Sakhrawi, Sellami e Bouassida(2022); Marapelli, Carie e Islam (2021)	4
MDMRE	Ali et al. (2023); Mahmood et al. (2020); Shahpar, Bardsiri e Bardsiri (2022)	3

MBRE	Abnane et al. (2023); Shukla e Kumar (2023);	2
EFFECT SIZE	Abnane et al. (2021); Hosni, Idri e Abran (2019)	2
MDAE	Ali et al. (2023)	1
MIBRE	Abnane et al. (2023)	1
LSD	Abnane et al. (2023)	1
AR	Cabral e Oliveira (2021)	1
ARLOG	Cabral e Oliveira (2021)	1
MRE ADJUSTED	Cabral e Oliveira (2021)	1
DELTA	Hosni, Idri e Abran (2021)	1
MAR	Rai, Kumar e Verma (2021)	1
SMAPE	Hosni, Idri e Abran (2019)	1
PCC	Hosni, Idri e Abran (2019)	1
CCC	Hosni, Idri e Abran (2019)	1
YY	Shukla e Kumar (2023)	1
AVERAGE EN	Charmanas, Mittas e Angelis (2020)	1
BMMRE	Shahpar, Bardsiri e Bardsiri (2022)	1
NÃO MENCIONA	Nosrati e Rahmani (2023)	1

Fonte: Elaborado pelos autores.

A Tabela 3 descreve as medidas de avaliação de precisão utilizadas nos estudos. É importante mencionar que a métrica de avaliação MAE foi amplamente utilizada nos estudos, seguidas pelas métricas PRED, SA, RMSE, MMRE, MSE, R2, MBRE e MDMRE. A métrica de avaliação MAE foi utilizada em 12 estudos (52,17%), às métricas PRED, SA e RMSE foram utilizadas em 7 estudos (30,43%), a MMRE foi utilizada em 6 estudos (26,08%), a MSE foi utilizada em 5 estudos (21,73%) e a métrica R2 foi utilizada em 4 estudos (17,39%). A Figura 5 mostra a distribuição das métricas de avaliação aplicadas nos estudos.

Figura 5: Distribuição das métricas de avaliação utilizadas no estudo



Fonte: Elaborada pelos autores.

D. QUAL A TÉCNICA DE VALIDAÇÃO MAIS ABORDADA NOS ESTUDOS?

A Tabela 4 detalha as técnicas de validação de aprendizado de máquina empregadas nos estudos. As técnicas de validação mais empregadas nos estudos foram o *Cross validation* (52,17%), *Leave-one-out* (43,47%), *Holdout* (8,69%) e *Bootstrap* (4,34%). Ademais, 4 estudos não mostram de forma clara qual a técnica de validação que foi empregada no projeto.

Tabela 4: Técnicas de validação utilizadas no estudo

Técnicas de validação	Estudos	Total
CROSS VALIDATION	Marco, Ahmad e Ahmad (2022); Goyal (2022); Ali et al. (2023); Labidi e Sakhrawi (2023); Rai, Kumar e Verma (2021); Shukla e Kumar (2023); Beesetti, Bilgaiyan e Mishra (2023); Rahman, Pandey e Ansari (2021); Sakhrawi, Sellami e Bouassida (2022); Sakhrawi, Sellami e Bouassida (2022); Shahpar, Bardsiri e Bardsiri (2022); Nosrati e Rahmani (2023)	12
LEAVE-ONE-OUT	Cabral e Oliveira (2021); Abnane et al. (2023); Hosni, Idri e Abran (2021); Shukla e Kumar (2023); Rahman, Pandey e Ansari (2021); Sakhrawi, Sellami e Bouassida (2022); Shukla e Kumar (2023) Abnane et al. (2021); Charmanas, Mittas e Angelis (2020); Hosni, Idri e Abran (2019)	10
NÃO MENCIONADO	Mahmood et al. (2020);Marapelli, Carie e Islam (2021); Srivastava, Sharma e Choudhary (2021); Kumar e Yeresime (2023)	4
HOLDOUT	Rahman, Pandey e Ansari (2021); Sakhrawi, Sellami e Bouassida (2022)	2
BOOTSTRAP	Nosrati e Rahmani (2023)	1

Fonte: Elaborada pelos autores.

E. QUAL A MELHOR TÉCNICA DE APRENDIZADO DE MÁQUINA EMPREGADA NOS ESTUDOS?

Escolher a melhor técnica de aprendizado de máquina para a estimativa da predição do esforço de equipe de *software* não é uma tarefa trivial. A literatura sugere que comitês de classificadores alcançam melhores resultados do que modelos autônomos. O quadro 3 exibe as técnicas de aprendizado de máquina usadas nos estudos.

Quadro 3: Técnica de aprendizado de máquina usada no estudo.

Técnicas	Estudo
O estudo aplicou o método de aprendizado por *ensemble AdaBoost* e *Random forest* (RF), o método de otimização bayesiana foi aplicado para determinar os hiperparâmetros do modelo.	Marco, Ahmad e Ahmad (2022)
O estudo propôs o uso de um comitê heterogêneo *Stacked* com Rede Neural e *Support Vector Machine*.	Goyal (2022)
O estudo utilizou modelos heterogêneos e de seleção dinâmica de conjuntos (DES), compostos por um conjunto de regressores	Cabral e Oliveira

selecionados de forma dinâmica pelos classificadores.	(2021)
O estudo desenvolveu 11 técnicas de construção de comitês de classificadores heterogêneos cujos membros são uma combinação de duas, três ou quatro técnicas de imputação únicas (SVRI, KNNI, DTI e EMI).	Abnane et al. (2023)
No estudo foi proposto modelos de estimativa autônomos, como Ponto de Caso de Uso, Julgamento Especializado (EJ) e Rede Neural Artificial.	Ali et al. (2023)
O estudo propôs um modelo de estimativa de esforço de *software* ensemble baseado em técnicas de *Use Case Points* (UCP) e julgamento de especialistas e *Case-Based Reasoning* (CBR).	Mahmood et al. (2020)
O estudo investigou três técnicas de seleção de recursos de filtro para verificar a capacidade preditiva de quatro técnicas: KNN, SVM, RNA e árvores de decisão.	Hosni, Idri e Abran (2021)
O estudo utilizou a técnica de conjunto *Stacking* denominada (StackSRTEE) que consistia em combinar três métodos individuais, Redes neurais (NN), Regressão de Vetor de Suporte (SVR) e Regressão de Árvore de Decisão (DTR).	Labidi e Sakhrawi (2023)
O estudo propôs um modelo de estimativa de esforço de *software ensemble* que utiliza as técnicas de (MLP) e (GLM).	Rai, Kumar e Verma (2021)
O estudo analisou 6 técnicas de aprendizado de máquina das quais 3 delas foram, prioritariamente, utilizadas como base para a construção dos conjuntos, sendo elas (DT , RF e SVR).	Shukla e Kumar (2023)
O estudo confrontou técnicas individuais e de conjunto como, Adaboost, Regressão Linear, *Random Forest*, *Gradient Boosting* e XGB, e propõem um conjunto usando *Voting*.	Beesetti, Bilgaiyan e Mishra (2023)
O estudo comparou um conjunto de aprendizado de máquina com um conjunto ponderado baseado em meta-heurística de HSBAs. Os algoritmos de regressão CART_R e FRBS_R e seus conjuntos ponderados com algoritmos de vaga-lumes, algoritmos genéticos e otimização de buraco negro são comparados com técnicas às aprendizado de máquina, *random forest, multilayerperceptron,* (SVM), *linear regression, bagging, stacking e voting*.	Rahman, Pandey e Ansari (2021)
O estudo fez uma comparação entre duas abordagens baseadas em aprendizado de máquina para estimar o EME de *software*: o modelo M5P (como um modelo individual) e o *stacking* como um método de conjunto que combina diferentes modelos de regressão (GBRegr, LinearSVR e RFR).	Sakhrawi, Sellami e Bouassida (2022)
O Estudo confrontou as técnicas individuais, KNN, SVR e DTR com as técnicas em conjunto: *Voting, Bagging e Boosting.*	Marapelli, Carie e Islam (2021)
O estudo fez uso de SVR (*Support Vector Regressor*) para prever o esforço necessário para aprimorar um *software* dentro do contexto de projetos Scrum.	Sakhrawi, Sellami e Bouassida (2022)
O estudo utilizou cinco diferentes técnicas de aprendizagem individual (LR, KNN, DT, SVR e MLP) que foram base para criação de técnicas	Shukla e Kumar

em conjunto das quais o modelo *Boost*-SVR se sobressaiu.	(2023)
O estudo desenvolveu e avaliou uma imputação de uma técnica de conjunto cujos membros são as quatro técnicas de imputação simples, KNN, EM, SVR e DT.	Abnane et al. (2021)
O estudo aplicou o modelo (EEABE), que combina o algoritmo genético (GA) com modelos ABE (Euclidiana, Manhattan, *Minkowski*, Máxima, *Akritean*, *Mahalanobis*). Os resultados do EEABE foram comparados com técnicas já utilizadas, como (GABE, BABE, PABE, OABE, MICBR, WGre-ABE, NABE, Ensemble, ANFIS-SBO e ANFIS-PSO).	Shahpar, Bardsiri e Bardsiri (2022)
Esse estudo abordou o desempenho de diversas técnicas individuais onde 18 delas são utilizadas na construção da tecnica em conjunto, dessas técnicas individuais há destaque para alunos CART e *Stepwise Regression* (SWReg).	Charmanas, Mittas e Angelis (2020)
O estudo desenvolveu uma estrutura homogênea de seleção de recursos de conjunto distribuído (HMDE-FS).	Nosrati e Rahmani (2023)
O estudo investigou se técnicas de analogia difusa de filtro único superaram conjuntos de analogia difusa de filtro.	Hosni, Idri e Abran (2019)
O estudo tratou de uma comparação entre dois modelos de regressão algorítmica, *Multiple Regression e Random Forest Regression.*	Srivastava, Sharma e Choudhary (2021)
O estudo investigou as capacidades e usos potenciais da utilização de Redes Neurais Artificiais (RNA) como ferramenta de previsão do esforço necessário para o desenvolvimento de *software*.	Kumar e Yeresime (2023)

Fonte: Elaborado pelos autores.

O Quadro 3 expõe às técnicas de aprendizado de máquina autônomas e aquelas que foram usadas em comitês de classificadores, é possível verificar que entre todas as técnicas autônomas utilizadas nos estudos o SVR (*Support Vector Regressor*) foi a técnica de aprendizado de máquina mais empregada com 34,78%. O *Randon Forest* e *Decision tree* empataram com 26,08% dos estudos. Quanto às técnicas de comitês de classificadores, o *Stacking* e *Boosting* empataram com 16,66% e a técnica *Voting* obteve 12,5%.

F. RESULTADO DA ANÁLISE DE QUALIDADE

Na análise de qualidade, cada um dos 23 estudos primários foi avaliado com base nas 8 questões, para a classificação de cada questão foi utilizada uma escala de positivos (1) e negativos (0). A linha representa o estudo primário e as colunas ‘Ql’ a ‘Q8’ representam as questões descritas anteriormente. A Tabela 5 apresenta o resultado da análise dos estudos primários.

Tabela 5: Análise da Qualidade dos Estudos Primários

Estudo	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Total
Rai, Kumar e Verma (2021)	1	1	1	1	1	1	1	1	8
Goyal (2022)	1	1	1	1	1	1	0	1	7
Cabral e Oliveira (2021)	1	1	1	1	1	1	0	1	7
Abnane et al. (2023)	1	1	1	1	1	1	0	1	7
Ali et al. (2023)	1	1	1	1	1	1	1	0	7
Hosni, Idri e Abran (2021)	1	1	1	1	1	1	0	1	7
Shukla e Kumar (2023)	1	1	1	1	1	1	1	0	7
Rahman, Pandey e Ansari (2021)	1	1	1	1	1	1	1	0	7
Sakhrawi, Sellami e Bouassida (2022)	1	1	1	1	1	1	1	0	7
Marapelli, Carie e Islam (2021)	1	1	1	1	1	1	1	0	7
Shukla e Kumar (2023)	1	1	1	1	1	1	0	1	7
Abnane et al. (2021)	1	1	1	1	1	1	0	1	7
Shahpar, Bardsiri e Bardsiri (2022)	1	1	1	1	1	1	0	1	7
Charmanas, Mittas e Angelis (2020)	1	1	1	1	1	1	0	1	7
Hosni, Idri e Abran (2019)	1	1	1	1	1	1	0	1	7
Marco, Ahmad e Ahmad (2022)	1	1	1	1	1	1	0	0	6
Beesetti, Bilgaiyan e Mishra (2023)	1	1	1	1	1	1	0	0	6
Labidi e Sakhrawi (2023)	1	1	1	1	0	1	0	0	5
Srivastava, Sharma e Choudhary (2021)	1	1	1	1	1	1	0	0	5
Mahmood et al. (2020)	1	1	1	1	1	0	0	0	4
Nosrati e Rahmani (2023)	1	1	1	1	1	0	0	0	4
Sakhrawi, Sellami e Bouassida (2022)	1	1	1	1	0	0	0	0	3
Kumar e Yeresime (2023)	1	1	1	1	0	0	0	0	3
Total	23	23	23	23	20	19	6	10	

Fonte: Elaborado pelos autores.

É possível observar na Tabela 5 que, os 23 estudos respondem às questões de 1 a 4, 3 estudos não apresenta de forma clara como os resultados da pesquisa foram validados, 4 estudos não apresentam de forma clara de como foi realizado a predição da estimativa de esforço de *software*, 17 estudos não apresentam as variáveis que geram maior impactos na predição da estimativa de esforço de *software* e apenas 10 estudos apresentam teste estatístico. Dentre todos os estudos destacam-se o estudo de Rai, Kumar e Verma (2021), com a maior soma de pontuação e os estudos Sakhrawi, Sellami e Bouassida (2022) e Kumar e Yeresime (2023), com a menor soma de pontuação.

5 CONCLUSÃO

A predição eficiente da estimativa de esforço de equipe de *software* é uma tarefa crucial para o sucesso de um projeto de *software*. As estimativas imprecisas podem causar insatisfação do cliente e, assim, diminuir a qualidade do produto entregue. Técnicas de aprendizado de máquina baseadas em comitês de classificadores podem melhorar a estimativa de tais esforços. O objetivo do trabalho foi realizar uma revisão sistemática da literatura sobre o uso de comitês de classificadores para predição da estimativa do esforço de equipe de *software*. Os resultados mostram que os *datasets* predominantemente utilizados nos 23 estudos primários utilizados na RSL foi o *datasets Desharnais* com 47.83% dos casos.

Dentre as ferramentas/frameworks utilizados nos estudos destacam-se as ferramentas *Weka* e *Google Colab*, utilizadas em 2 estudos. *Cross validation* foi a técnica de validação mais empregada nos estudos. Sobre as métricas de avaliação da estimativa de esforço de equipe de *software* é importante mencionar que a métrica de avaliação MAE foi amplamente utilizada nos estudos, seguidas pelas métricas PRED, SA, RMSE, MMRE, MSE, R2, MBRE e MDMRE.

Na avaliação da qualidade dos estudos, o estudo de Rai, Kumar e Verma (2021) obteve a maior soma de pontuação. Cada técnica apresentada nos estudos possui seus pontos positivos e negativos, portanto não é possível afirmar qual é a melhor técnica de aprendizado de máquina para estimativa da predição do esforço de equipe de *software*.

Por meio do desenvolvimento da RSL, foi possível visualizar as lacunas presentes nas pesquisas que envolvem o uso de comitês de classificadores na estimativa de esforço de software, tais como a falta de informação acerca dos *frameworks*/ferramentas mais utilizadas, como também a escassa quantidade de estudos que mencionam testes estatísticos utilizados. sendo assim, para trabalhos futuros identificamos a necessidade de utilização de algoritmos de otimização para as fases de treinamento e teste dos comitês de classificadores, bem como maior detalhamento das ferramentas utilizadas na construção das técnicas.

REFERÊNCIAS

ABNANE, Ibtissam et al. Evaluating ensemble imputation in software effort estimation. **Empirical Software Engineering,** v. 28, n. 2, p. 56, 2023.

ABNANE, Ibtissam et al. Heterogeneous ensemble imputation for software development effort estimation. In: **Proceedings of the 17th International Conference on Predictive Models and Data Analytics in Software Engineering**. 2021. p. 1-10.

ALI, Syed Sarmad et al. Heterogeneous Ensemble Model to Optimize Software Effort Estimation Accuracy. **IEEE Access,** v. 11, p. 27759-27792, 2023.

ARAÚJO, Débora da Conceição. Avaliação de comitês com classificadores tradicionais e profundos para análise de sentimentos. 2019. Dissertação de Mestrado. Universidade Federal de Pernambuco.

BEESETTI, Kiran Kumar; BILGAIYAN, Saurabh; MISHRA, Bhabani Shankar Prasad. Software Effort Estimation through Ensembling of Base Models in Machine Learning using a Voting Estimator. **International Journal of Advanced Computer Science and Applications,** v. 14, n. 2, 2023.

CABRAL, Jose Thiago H. de A.; OLIVEIRA, Adriano LI. Ensemble Effort Estimation using dynamic selection. **Journal of Systems and Software**, v. 175, p. 110904, 2021.

CHARMANAS, Konstantinos; MITTAS, Nikolaos; ANGELIS, Lefteris. Ensemble Software Development Effort Estimation Using Data Envelopment Analysis. In: **Proceedings of the 24th Pan-Hellenic Conference on Informatics.** 2020. p. 202-207.

GOYAL, Somya. Effective software effort estimation using heterogenous stacked ensemble. In: **2022 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES)**. IEEE, 2022. p. 584-588.

HAI, Vo Van et al. A new approach to calibrating functional complexity weight in software development effort estimation. **Computers**, v. 11, n. 2, p. 15, 2022.

HOSNI, Mohamed; IDRI, Ali; ABRAN, Alain. Evaluating filter fuzzy analogy homogenous ensembles for software development effort estimation. **Journal of Software: Evolution and Process,** v. 31, n. 2, p. e2117, 2019.

HOSNI, Mohamed; IDRI, Ali; ABRAN, Alain. On the value of filter feature selection techniques in homogeneous ensembles effort estimation. **Journal of Software: Evolution and Process,** v. 33, n. 6, p. e2343, 2021.

IDRI, Ali; HOSNI, Mohamed; ABRAN, Alain. Systematic literature review of ensemble effort estimation. **Journal of Systems and Software**, v. 118, p. 151-175, 2016.

JADHAV, Anil; KAUR, Mandeep; AKTER, Farzana. Evolution of software development effort and cost estimation techniques: five decades study using automated text mining approach. **Mathematical Problems in Engineering**, v. 2022, p. 1-17, 2022.

JADHAV, Akshay et al. Effective Software Effort Estimation enabling Digital Transformation. **IEEE Access**, 2023.

KITCHENHAM, Barbara et al. Systematic literature reviews in software engineering–a systematic literature review. Information and software technology, v. 51, n. 1, p. 7-15, 2009.

KUMAR, B. R.; YERESIME, S. Software Effort Estimation using ANN (Back Propagation). In: **7º Congresso Internacional de Metodologias de Computação e Comunicação (ICCMC),** 2023. IEEE, p. 1-2.

LABIDI, Taher; SAKHRAWI, Zaineb. On the value of parameter tuning in stacking ensemble model for software regression test effort estimation. **The Journal of Supercomputing**, p. 1-23, 2023.

MAHMOOD, Yasir et al. Improving estimation accuracy prediction of software development effort: A proposed ensemble model. In: **2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)**. IEEE, 2020. p. 1-6.

MAHMOOD, Yasir et al. Software effort estimation accuracy prediction of machine learning techniques: A systematic performance evaluation. **Software: Practice and experience**, v. 52, n. 1, p. 39-65, 2022.

MARAPELLI, Bhaskar; CARIE, Anil; ISLAM, Sardar MN. Software effort estimation with use case points using ensemble machine learning models. In: **2021 International Conference on Electrical, Computer and Energy Technologies (ICECET)**. IEEE, 2021. p. 1-6.

MARCO, Robert; AHMAD, Sakinah Sharifah Syed; AHMAD, Sabrina. Bayesian hyperparameter optimization and Ensemble Learning for Machine Learning Models on software effort estimation. **International Journal of Advanced Computer Science and Applications,** v. 13, n. 3, 2022.

NOSRATI, Vahid; RAHMANI, Mohsen. HMDE‐FS: A homogeneous distributed ensemble feature selection framework based on resampling with/without replacement. **Concurrency and Computation: Practice and Experience,** v. 35, n. 7, p. e7613, 2023.

RAI, Prerana; KUMAR, Shishir; VERMA, Dinesh Kumar. Prediction of software effort in the early stage of software development: a hybrid model. **IEEE Canadian Journal of Electrical and Computer Engineering**, v. 44, n. 3, p. 376-383, 2021.

RHMANN, Wasiur; PANDEY, Babita; ANSARI, Gufran Ahmad. Software effort estimation using ensemble of hybrid search-based algorithms based on metaheuristic algorithms. **Innovations in Systems and Software Engineering**, p. 1-11, 2021.

SAKHRAWI, Zaineb; SELLAMI, Asma; BOUASSIDA, Nadia. Software enhancement effort estimation using correlation-based feature selection and stacking ensemble method. **Cluster Computing**, v. 25, n. 4, p. 2779-2792, 2022.

SAKHRAWI, Zaineb; SELLAMI, Asma; BOUASSIDA, Nadia. Support vector regression for enhancement effort prediction of Scrum projects from COSMIC functional size. **Innovations in Systems and Software Engineering**, v. 18, n. 1, p. 137-153, 2022.

SHAHPAR, Zahra; BARDSIRI, Vahid Khatibi; BARDSIRI, Amid Khatibi. An evolutionary ensemble analogy‐based software effort estimation. **Software: Practice and Experience**, v. 52, n. 4, p. 929-946, 2022.

SHUKLA, Suyash; KUMAR, Sandeep. Self-Adaptive Ensemble-based Approach for Software Effort Estimation. In: **2023 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)**. IEEE, 2023. p. 581-592.

SHUKLA, Suyash; KUMAR, Sandeep. Towards ensemble-based use case point prediction. **Software Quality Journal**, p. 1-22, 2023.

SRIVASTAVA, Devesh Kumar; SHARMA, Akhilesh Kumar; CHOUDHARY, Deevesh. Software Development Effort Estimation Using Machine Learning Techniques: Multi-linear Regression versus Random Forest. In: **2021 International Conference on Computing, Communication and Green Engineering (CCGE)**. IEEE, 2021. p. 1-5.

SREEKANTH, N. et al. Evaluation of estimation in software development using deep learning-modified neural network. **Applied Nanoscience**, v. 13, n. 3, p. 2405-2417, 2023.

SURESH KUMAR, P. et al. A pragmatic ensemble learning approach for effective software effort estimation. **Innovations in Systems and Software Engineering**, v. 18, n. 2, p. 283-299, 2022.