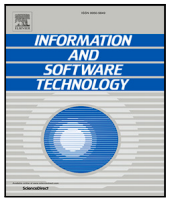
Information and Software Technology 148 (2022) 106912

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof



# An empirical study of emoji use in software development communication

Shiyue Rong [a], Weisheng Wang [a], Umme Ayda Mannan [b], Eduardo Santana de Almeida [c], Shurui Zhou [d], Iftekhar Ahmed [a,*]

[a] University of California, Irvine, USA
[b] Oregon State University, USA
[c] Federal University of Bahia, Brazil
[d] University of Toronto, Canada

ARTICLE INFO

*Keywords:*
Emoji
Software engineering
Sentiment analysis
Empirical analysis

ABSTRACT

**Context:** Similar to social media platforms, people use emojis in software development related communication to enrich the context and convey additional emotion. With the increasing emoji use in software development-related communication, it has become important to understand why software developers are using emojis and their impact.

**Objective:** Gaining a deeper understanding is essential because the intention of emoji usage might be affected by the demographics and experience of developers; also, frequency and the distribution of emoji usage might change depending on the activity, stage of the development, and nature of the conversation, etc.

**Methods:** We present a large-scale empirical study on the intention of emoji usage conducted on 2,712 Open Source Software (OSS) projects. We build a machine learning model to automate classifying the intentions behind emoji usage in 39,980 posts. We also surveyed 60 open-source software developers from 17 countries to understand developers' perceptions of why and when emojis are used.

**Results:** Our results show that we can classify the intention of emoji usage with high accuracy (AUC of 0.97). In addition, the results indicate that developers use emoji for varying intentions, and emoji usage intention changes throughout a conversation.

**Conclusion:** Our study opens a new avenue in Software Engineering research related to automatically identifying the intention of the emoji use that can help improve the communication efficiency and help project maintainers monitor and ensure the quality of communication. Another thread of future research could look into what intentions of emoji usage or what kind of emojis are more likely to attract users and how that is associated with emoji usage diffusion in different levels (threads, projects, etc.)

## 1. Introduction

Communication between humans is continually changing and adapting to social trends, lifestyles, and technology. Since language responds to social change and attitudes, its forms and usage also evolve according to its users' needs and the tools they use for communication [1]. One such tool is smartphones, whose widespread use has introduced applications of embedding emojis in conversation. Emojis are small digital images or icons used to express an idea, emotion, object, etc. [2], and have become one of the quick means of expressing sentiments, enriching the context [3–6] and is widely used in various social media platforms.

According to Emojipedia [7], an analysis of over 68 million tweets shows that nearly one in five tweets (19.04%) contain at least one emoji. Data from Facebook also shows an average of 5 billion emojis sent each day on Messenger [8]. In addition, the use of emojis increases engagement by 48% on Instagram, where posts that use emojis on Instagram have an interaction rate of 2.21% while posts without emojis have 1.77% [9].

Emojis are becoming more prevalent in software development both as part of programming languages and in software development-related communication. For example, Emojicode [10] was developed as the first programming language consisting of emojis. Programming languages, such as Python [11] and JavaScript [12], also support embedding emojis in source code. GitHub users have also started using emoji, and recent studies have shown a considerable proportion of the emojis usage on GitHub [13]. As a result of such widespread use, to

* Corresponding author.
*E-mail addresses:* shiyuer@uci.edu (S. Rong), weishew@uci.edu (W. Wang), mannanu@oregonstate.edu (U.A. Mannan), esa@rise.com.br (E.S. de Almeida), shuruiz@ece.utoronto.ca (S. Zhou), iftekha@uci.edu (I. Ahmed).

https://doi.org/10.1016/j.infsof.2022.106912
Received 6 October 2021; Received in revised form 23 March 2022; Accepted 28 March 2022
Available online 11 April 2022

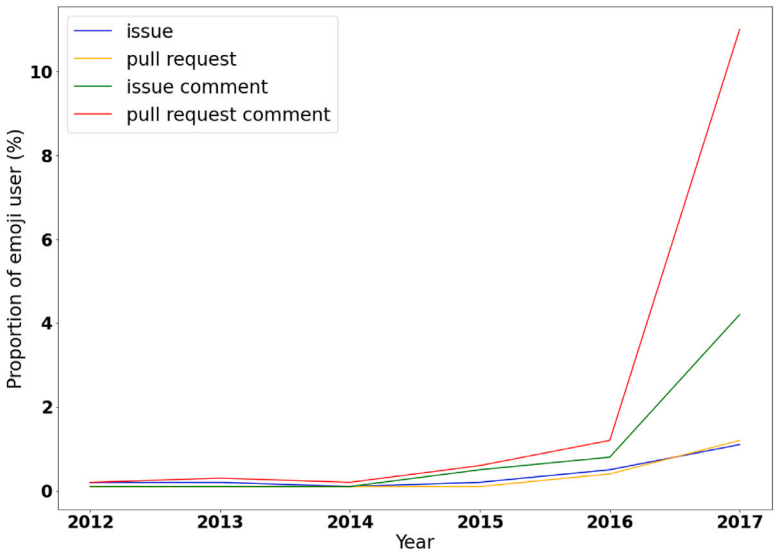



**Fig. 1.** Proportion of emoji users in GitHub conversations with a sharp increase after the release of the emoji reaction feature in 2016, from Lu et al. [13].

standardize emoji use and enhance effective communication, guidelines for using emojis in commit messages have been proposed [14–16]. For example, 🚀 represents deployment, and 💄 means adding or updating the UI and style according to this guideline [15]. Emojis are not only used to convey meanings based on their appearance but also for expressing context-dependent meanings. For example, the rocket emoji used in "wow you are fast 🚀" can be inferred as representing a sentiment, and the usage in "Think we are done, let us deploy 🚀" refers to the meaning mentioned in the guideline [15].

Due to the prevalence of emoji usage shown in Fig. 1, researchers have recently started investigating emoji use in software development tasks. For example, Lu et al. [13] reported a growing trend of emoji usage on GitHub and found that sentimental usage (e.g., adding emotions, strengthening an emotion, etc.) is the primary intention of using emoji in issues, pull requests, and comments. For example, "😃 Thank you for reviewing my code". is used for expressing gratitude. When using emojis, there are also non-sentimental intentions (e.g., drawing attention, replacing a phrase, etc.) For example, "🚨 Please review the contributing guidelines before creating and pushing any content". is used for drawing attention.[1] Additionally, researchers have started looking into techniques for sentiment analysis of emoji in the context of Software Engineering (SE) tasks by checking the sentimental polarity (positive, negative, neutral) [17] and emotions (e.g., joy, sadness, angry) [18].

Despite the interest from researchers, our understanding regarding how software developers are using emojis and their impact on software development communication is still limited. Gaining a deeper understanding is essential because emoji usage is rapidly increasing in SE-relevant artifacts. We need mechanisms to facilitate the appropriate emoji usage and understand any negative impact that may stem from such wide adoption. Moreover, the intention of emoji usage might be affected by the demographics and experience of developers [19]. Prior work shows that the role (core vs. non-core) of developers in a project is significantly related to their activities (i.e., number of messages committed, amount of conversations involved) [20]. Since emojis are used in these activities, we posit that there will be a difference in emoji usage based on a developer's role.

The frequency and the distribution of emoji usage intentions may vary depending on the activity, stage of development, and nature of the conversation as people's intention and temper differ between the time when they start a conversation and the time when they want to end a conversation [21].

Therefore, the frequency and intention of emoji usage can be used to retrieve the overall emotional state of a project [22] by tracking the intentions of the majority of participants. Our proposed technique can be used for monitoring improper use of emojis in a project (e.g. The comment in this issue, "your comment and 'eyes emoji' seems rather rude and uncooperative")[2] and help project maintainers in ensuring a healthy and welcoming environment.

Following this line of research, our exploratory study aims to fill the gap in our understanding by investigating emoji usage in a broader context, including both sentimental and non-sentimental intentions. We started by collecting a large-scale dataset (39,980 posts) with emojis from GitHub. Since manually labeling the intention of each emoji usage is unrealistic, we built a machine learning classifier to automate the process. We collected conversations with emojis from 2,712 open-source projects available from GHTorrent [23], and then performed this labeling process. Finally, we studied whether emojis have any effect while communicating and surveyed 60 open-source software developers from 17 countries to understand developers' perceptions of why and when emojis are used and who is using them.

Specifically, this paper addresses the following research questions:

**RQ1: How well can we identify the intentions of emoji usage using machine learning?**

**RQ2: What intentions do developers have while using emojis during conversations?**

**RQ3: When do developers use emojis the most during the conversation?**

**RQ4: Do core and non-core developers use emojis differently?**

The contributions of this paper are listed below:

- We present the first study investigating the applicability of machine learning techniques for identifying the intention of using an emoji.
- We present the result showing the role of developers in a project (core or non-core contributor) does not impact the number and the intention of emoji usages.

[1] https://github.com/gazaskygeeks/Fundamentals-course/blob/master/README.md

[2] https://github.com/andig/homebridge-fritz/issues/124

- We present the results of a survey of 60 software developers expressing their opinion regarding why, and where emoji is used in software development communication.
- Based on our results, we outline implications for developers and researchers.

review of prior research efforts. In Section 3, we present our methodology, the demographics of our corpus, machine learning classifier to classify emoji use intentions, data collection process, and surveying developers for answering our intended research questions. In Section 4, we present our findings. Section 5 discusses the results and outlines implications for developers and researchers. Section 6 discusses the limitations that could affect the validity of our study. Section 7 concludes with a summary of the key findings and future work.

## 2. Related work

Our study is motivated by the following two main streams of research: Emoji usage in SE and emojis in SE specific sentiment analysis.

Recently, researchers have started to analyze emoji usage in software development platforms. Claes et al. [24] investigated the use of emoji in open source software development for Apache and Mozilla. They found that Mozilla developers use more emoji than Apache developers and emoji usage could also help to detect developers' mental health. Lu et al. [13] analyzed the emoji usage on GitHub and found that emoji usage among GitHub users is growing. They also found that emojis are not only used to express sentiment in issues, pull requests, and commits, but also used to emphasize important contents in README files. Motivated by their work, instead of manually identifying sentiments and intentions, our study aims to facilitate the emoji usage intention analysis by automating this process. Extending the dataset from Lu et al. we also study "GitHub Discussion", a newly introduced communication channel on GitHub [25].

Given emojis' growing popularity, researchers have also started incorporating emojis in SE-specific sentiment analysis. Though sentiment analysis in SE has been a research interest for a while [26–35], most of the tools used in these studies are trained on non-SE texts. Several studies have reported limitations of these off-the-shelf sentiment analysis tools in analyzing sentiment in SE text [35–39]. To mitigate these shortcomings, researchers have come up with SE-specific sentiment analysis tools (SentiStrength-SE [40], SentiCR [39], Senti4SD [41]) by leveraging SE-related texts from different code review platforms. However, only using text cannot fully capture developers' sentiment as they also use emojis to express their emotions. As a step towards incorporating emojis into SE-specific sentiment analysis, Chen et al. proposed a tool *SEntiMoji* [17,18] that combines both text and emoji and gets a higher prediction accuracy compared to all the other existing tools. Their first study [17] categorized the sentimental polarity of emoji usage into three categories, positive, negative, and neutral. In their follow-up study [18], they improved their *SEntiMoji* model to detect the exact sentiments (Love, Anger, Joy, Sadness, etc.). Different from their categorization, we investigate the intention of emoji usage not only for sentimental expressions but also for non-sentimental expressions such as drawing attention and object representation.

Though current research has started to investigate emoji usage in software development activities and SE-specific sentiment analysis, there are still gaps in understanding the intentions behind emoji use in software development. Thus far, no work has investigated how emoji usage varies depending upon the phase of communication, developer experience level, etc.

In our study, we aim to fill these gaps.

**Table 1**
Dataset summary.

| | #issue comments | #pull request comment | #commit message | #discussion comment |
|---|---|---|---|---|
| Non-emoji | 5,183,558 | 26,991,345 | 4,006,138 | 4,696 |
| Emoji | 5,908 | 30,046 | 3,778 | 248 |
| Total | 5,189,466 | 27,021,391 | 4,009,916 | 4,944 |

## 3. Methodology

This study aims to understand why and where emojis are used in SE projects and who tends to use them. In order to do so, we focus on the conversations in open-source projects with emoji usage and start by building four different machine learning models (RQ1) for identifying the intention of emoji usage (RQ2). Next, we apply the best-performing classifier to answer the remaining research questions (RQ3-4). In the following subsections, we describe the pipeline in detail.

### 3.1. Data collection

Our overarching goal is to understand emoji usage intentions in the conversations among developers in open-source projects. For this, we first learned from the historical data of emoji usage by collecting all the conversational posts from January 2015 to June 2019 from GHTorrent [23], as the proportion of emoji usage on GitHub has increased since 2015 [13]. Specifically, our dataset contains issue comments, pull request comments, and commit messages, as these are the places where most SE conversations occur. Additionally, we included conversations from a newly introduced GitHub feature—"Discussion" [25], where developers from the same organization form teams to discuss topics not particularly related to pull requests or issues. Though discussions were used by only a small number of projects within some organizations, we still wanted to analyze their emoji usage. Since discussion comments were not available in GHTorrent, we crawled discussion comments using the Python library *Scrapy* [42].

After collecting all conversational posts, we filtered out duplicates, and non English texts. Next, we used Python's emoji library [43] to identify the posts with emojis. Posts with different types of emoji are split into multiple posts, each was considered as an unique post with the same text content but different kind of emoji embedded. For example, "⚠️ Few simple typos:-lunch ➡️ launch -Kgelseymentioned ➡️ Kelsey mentioned -kibernetes ➡️ kubernetes" will be split into "⚠️ Few simple typos:-lunch launch -Kgelseymentioned Kelsey mentioned -kibernetes kubernetes", and "Few simple typos:-lunch ➡️ launch -Kgelseymentioned ➡️ Kelsey mentioned -kibernetes ➡️ kubernetes". We did not split the posts where the same emoji is used multiple times. We posit that different emojis used in a sentence convey different intentions, while the same emoji used multiple times convey a similar intention for that sentence. Therefore, we split the statement with different emoji usages so that the predicted intention of one emoji will not be affected by the position or embedded information of another emoji used in this same statement. Since multiple occurrences of the same emoji are not prone to this, we did not split the statement with the same emoji. Our final dataset contained 2,712 open-source projects and 39,980 posts with the emoji usage, shown in Table 1.

### 3.2. Building the intention classifier

To answer our research questions, we needed to identify the developer's intentions of using emojis. As manual detection of intentions was not a practical option for 39,980 emoji posts, we used machine learning techniques to automate the step of detecting the developer's intention behind emoji usage.

### 3.2.1. Training data labeling

Before training the classifiers, we needed to prepare the training dataset. For this, we selected SentiMoji [17]'s Github dataset [44] with 1,690 comment data. We checked and removed any duplicates between SentiMoji's Github dataset and our GHTorrent dataset. We started with the seven intention labels[3] of emoji usage proposed by Lu et al. [13] where they identified the categories using manual analysis of the GitHub dataset, and no specific was metric used for this purpose. However, through our manual analysis of the intention categories, we found that there were some overlapping categories and ambiguities due to naming.

Our analysis revealed that *Statement enriching* proposed by Lu et al. [13] could be divided into *Object Representation* and *Object Replacement*, since the original definition includes both the usage of replacing the context and illustrating the context. Lu et al. also had a category named"Emoji" which turns out to be overlapped with *Object Replacement*. We also found that *Sentimental usage* category had two subcategories which would be difficult for the machine learning classifier to learn.

To avoid any complications due to overlapping categories, we decided to develop our own categories by adapting the categories proposed by Lu et al. [13]. Our manual classification was conducted using the "negotiated agreement" method [45] by three researchers[4] in two phases. In the first phase, we started by randomly selecting 169 (10%) emoji posts out of 1,690 posts which is the standard practice in negotiated agreement. Three of the authors individually labeled these 169 posts into either existing categories or came up with a new intention labeled by themselves if the post does not match the existing ones. We calculated the inter-rater reliability using Fleiss' Kappa [46] after the first phase and found a Kappa value of 0.69. Fleiss' Kappa is a statistic value that assesses the degree of agreement between the codes assigned by three or more researchers working independently on the same sample. Values of Fleiss' kappa fall between 0 and 1, where 0 indicates poor agreement and 1 indicates perfect agreement. According to the thresholds, the kappa value of 0.69 indicates a substantial agreement [47] between the researchers.

In the second phase, the researchers continued the negotiation process to come up with the finalized intention categories. In this phase modifications of categories included merging, renaming and removing categories. We merged *Atmosphere adjustment* [13] with *Sentimental Strengthening* as adjusting tone serves the same purpose as expressing sentiments. We renamed *Content emphasis* [13] to *Visual Enhancement* because *Visual Enhancement* better captures the definition of category and also to avoid any confusion with *Content Organization*. We split *Sentimental usage* into two categories, named *Sentimental Addition* and *Sentimental Strengthening*. We removed *Unintentional usage* since there is no way to detect if an emoji was used accidentally or not.

The final six categories of intentions are as following.

(1) **Content Organization (CO)** means adding emojis to allocate and organize different content, and it is usually used for improving readability. For example, when multiple tasks are addressed, emojis such as ✅, ➡️, and ✏️ can be used for indicating checklists.

(2) **Object Replacement (O. Repl)** means directly replacing the object or word with the emoji which has the same meaning. For example, in the issue comment "Can you ✂️ the old default part", the emoji here replaces the word such as "cut" and "remove" in the message.

(3) **Object Representation (O. Repr)** means when a statement already includes the object or word, however, the emoji is still added for representing that word. For example, in the issue comment "Ahh right. Fixup below! ⬇️", the emoji here represents the word "below" even though the word is present.

(4) **Sentimental Addition (SA)** means adding emojis to express emotions in the absence of words expressing the emotion. For example, in the pull request comment "now it should work. seems need to test 2.0.2 rc on php 5.3 too 😕", the attitude is expressed through the 😕 emoji.

(5) **Sentimental Strengthening (SS)** means adding emojis to make the post more expressive when the emotion is already expressed. For example, in the pull request comment "awesome!! they both work 😃", the 😃 emoji used here further strengthens the word "awesome" in the text.

(6) **Visual Enhancement (VE)** means adding emojis for drawing attention using the visual appearance of an emoji. For example, ⚠️, ✨, and ❗ can be used in a post even though they are not semantically or sentimentally related to the message but contributing their graphic appearance to the post.

Once the agreement was reached and the set of intention labels are finalized, the first and second author then manually classified the rest of the remaining (1,521) training data. Table 2 shows the distribution of manually classified training data in each category.

**Table 2**
Distribution of manually classified sample across different labels.

| Category | Count |
|---|---|
| Content Organization (CO) | 229 |
| Object Replacement (O. Repl) | 191 |
| Object Representation (O. Repr) | 242 |
| Sentimental Addition (SA) | 262 |
| Sentimental Strengthening (SS) | 455 |
| Visual Enhancement (VE) | 311 |

Because of the imbalance in category distribution of the training data and the high cost of incorrectly classifying minority example, we duplicated the samples for all categories except for "Sentimental Strengthening" by using the Synthetic Minority Over-sampling Technique (SMOTE) [48] to reduce the class-imbalance. The resulting distribution has a count of 455 for each intention category, which sums up to a total of 2,730 training data.

### 3.2.2. Feature selection

There were no prior classifiers in existing literature built for identifying intention of emoji usage, hence we designed our own feature set. We explored various features which includes emoji definitions, emoji position in a sentence, emoji usage frequency, and emoji polarity etc. In total, we selected eight features. Details of these features are presented in Table 3 and described below.

**Frequency of individual emoji**. This feature counts how many times the same emoji is used in one statement after splitting. Our manual analysis revealed that the same emoji might be used multiple times to clarify the intention or for drawing attention. Hence, we posited that the frequency of an emoji in a sentence can help to categorize the intention. For instance, "🚑🚑 Class constructor fixed". and "🚨 [Warning] 🚨 gcc version conflicts" will both get the value 2 for this feature. For posts with multiple different emojis, they are split into multiple posts, and each of them is considered as a unique post with the same text but with a different kind of emoji embedded, as explained in Section 3.1.

**Position of emoji**. We calculated the relative position of an emoji in a sentence by dividing the index of the emoji by the length of the whole sentence. If more than one emoji of the same type is present, we calculated the average of relative positions. The value of this feature ranges from 1 to 100. We selected this feature because our manual analysis revealed that position of the emoji conveys information regarding the intention. For example, in the case of SA and SS, emojis are primarily used at the end of the sentence (values roughly range from 95 – 100).

[3] Sentimental usage, Statement enriching, Content organization, Content emphasis, Atmosphere adjustment, Unintentional usage, Emoji

[4] Three researchers are Ph.D. students with three to five years of real-world experience in open-source software development.

**Table 3**
Feature summary.

| Feature | Description |
|---|---|
| Frequency of individual emoji | Counts how many times the same emoji is used in one statement |
| Position of emoji | Calculates the relative position of an emoji in a sentence by dividing the index of the emoji by the length of the whole sentence |
| Emoticons | Conveys whether the emoji belongs to the "Emoticons" category, presented by boolean value |
| Rate of emoji usage | Contains three features summed up to one: portion of positive, negative, and neutral usages of one emoji |
| Similarity between emoji definition and statement | Calculates the similarity score between emoji definition and text in the statement by iterating each words from definition and statement |
| Polarity of statement | Identifies the emotion of the text within statement itself, calculated using the NLP library |

On the other hand, in VE, emojis are primarily used at the beginning of the sentence (values roughly range from 1 – 5). And for CO, the emojis are mostly used in the middle of a sentence (values range around 50).

**Emoticons**. Our manual analysis revealed that emojis belonging to the "Emoticons" category conveys information regarding the emotion of the emoji itself, for example, 😃 and 😞. We used the categorization provided by Novak et al. [6] to identify if an emoji belongs to the "Emoticons" category or not and used a boolean value (0 or 1) to represent this feature.

**Rate of emoji usage**. As emojis themselves can have positive, negative, and neutral emotions, we believe such attribute is related to the intention of using them. Novak et al. [6] after manually labeling 1.6 million tweets, calculated the rate at which each emoji is used for expressing a positive, negative or neutral emotion. For each emoji in our dataset, we looked up the rate in Novak et al. [6] dataset and use each of the three rates as a separate feature. These three features sum up to one for an emoji. For example, "makefile test failed 😞" will get value 0.07, 0.75, and 0.18, if number of usage is 7 (positive), 75 (negative), and 18 (neutral) from Novak's dataset, respectively.

**Similarity between emoji definition and statement**. Since emojis can be used for representing objects, understanding how the meaning of emoji is related to the context of the sentence itself is important. For each emoji, we checked if the definition of the emoji extracted from their Common Locale Data Repository (CLDR) short name [49] was similar to any of the words present in the sentence. To calculate the similarity score between CLDR short name and each word present in the sentence, we use NLTK's wordnet package [50]. The value of this feature ranges between 0 to 1. For example, as the CLDR short name of "⬇" is "down arrow", "Detailed fixing tips are listed below.⬇" will get 0.9 based on the score between "down" and "below", which is the word pair with the highest similarity value.

**Polarity of statement**. This feature was primarily used for identifying if the text itself has any emotion, so that it allowed us to identify whether the emoji is adding emotion or strengthening an already expressed emotion in the context. We calculated the polarity using *TextBlob* [51], a library for natural language processing (NLP) tasks, which the scores ranged from −1 (negative) to 1 (positive). We also looked into SentiCR [39] and decided not to use it since SentiCR used code review comments from Gerrit for training purposes. Authors of that paper mentioned that their tool might not be appropriate for other communication channels, such as pull requests in GitHub due to the difference in vocabulary and communication platform (GitHub vs. Gerrit). Since we analyzed four different data types with different communication purposes, we chose TextBlob, as a more general-purpose solution.

Since the features have different range of values, we applied the feature scaling technique [52] to normalize each features value between zero and one. Using the following equation (Eq. (1)). This helps to improve the performance of machine learning models:

$$X_{scaled} = \frac{X - X_{min}}{X_{max} + X_{min}} \tag{1}$$

### 3.2.3. Machine learning classifier

Using the manually classified dataset, we trained four different machine learning classifiers: Decision Tree (DT) [53], Logistic Regression (LR) [54], Random Forest (RF) [55], and Support Vector Machine (SVM) [56]. Since there are no prior models developed for emoji usage classification, we selected these models as they performed well for the sentiment classification task, which is relatively similar to the task at hand [57–60].

**Decision Tree Classifier (DT):** We used Decision Tree Classifier because it outputs easily interpretive rules and feature importance that measure the predictive power of the feature. We built the model using the *Gini Impurity* criterion which measures the likelihood of incorrectly labeling a randomly chosen variable, if the variable is randomly labeled according to the distribution of labels in the dataset [61]. As *Gini Impurity* goes down, the probability of miss classification also goes down.

**Multinomial Logistic Regression (LR):** For the Multinomial Logistic Regression, we used Limited-memory BFGS (L-BFGS) [62] algorithm for parameter estimation. To avoid overfitting, we applied both L1 and L2 regularization.

**Random Forest (RF):** Because a single decision tree tends to overfit, we used Random Forest model to avoid such overfitting problems. By using Random Forest, we could reduce the variance of error so that we could get more reliable classification results. Same as building the Decision Tree Classifier, we used the *Gini Impurity* criterion to measure the quality of each split within the Tree models. The maximum number of features for building each Tree model equals to the square root of the total number of features.

**Support Vector Machine (SVM):** Based on the assumption that our intention labels could be linearly separated across the features, we experimented with using an SVM for classification. For the kernel type, we selected the standard *Radial Bases Function* (RBF) kernel. To set a scaled kernel coefficient *gamma*, we applied the formula:

$$gamma = \frac{1}{Nf \times VarData} \tag{2}$$

Where *Nf* is the number of features and *VarData* is the variance of data across all features

We used Python Scikit-learn library [63] to implement the classifiers. To take care of unseen test set problems, we used the hold-out method which 20% of the original data were used for testing, and then we performed 10-fold cross validation on the remaining training data to train and evaluate the classifiers [64]. This validation approach randomly divide the manually classified dataset into 10 groups of equal size. The first group is treated as a validation set, and the classifier is fit on the remaining 9 groups. The mean of the 10 executions is used as an estimation of classifier's accuracy. 10-fold cross validation has been recommended in the field of applied machine learning [65]. We performed hyper-parameter optimization using randomized search [66] for all four classifiers.

#### 3.2.4. Evaluation

We report the standard precision, recall, F1-score, and Area Under the Receiver Operating Characteristic Curve (AUC) to assess the performance of the classifiers. Our model computes the probability distribution of total six intention categories and then picked the highest one as the result. We used AUC instead of accuracy because accuracy does not relate the prior probability distribution of the classes due to a cut-off thresh hold while AUC relates the true-positive rate to the false-positive rate of all prior distributions which is a better indicator of the overall performance [67]. Also, AUC is a better measure of classifier performance because it is not biased by the size of test data. Moreover, AUC provides a "broader" view of the performance of the classifier since both sensitivity and specificity for all threshold levels are incorporated in calculating AUC. Other works related to prediction have used AUC for comparison purposes [68–71]. We listed the formula for measuring precision, recall, and F1-score below. The AUC curve is created by plotting the recall against the false positive rate (FPR) at various threshold settings. We listed the formula of FPR also.

- **Precision**: The portion of correct identifications from the predicted labels of a particular category.

$$precision = \frac{TP}{TP + FP} \tag{3}$$

- **Recall**: The portion of correct identifications from the actual (original) labels of a particular category.

$$recall = \frac{TP}{TP + FN} \tag{4}$$

- **F1 score**: The harmonic mean of precision and recall

$$F_1 = \frac{2TP}{2TP + FP + FN} \tag{5}$$

- **False positive rate**: A measure of the ratio of the number of wrongly categorized negative events and the total number of actual negative events.

$$FPR = \frac{FP}{FP + TN} \tag{6}$$

### 3.3. Communication phase identification

To answer RQ3 (When do developers use emojis the most during the conversation?), we needed to identify in which phase of the communication developers used emoji. We analyzed the posts in issues, pull requests, and the GitHub discussion. We did not consider commits since there are no back and forth conversations in commits. We started by sorting the posts in a conversation in according to the order of appearance. Then we calculated what percentage the posts with emoji belongs to with respect to all the other posts. We did this by dividing the chronological rank of the post by the total number of posts in that conversation. In general, conversations of solving problem involve three distinct phases (i.e. pre-engagement, engagement, and disengagement) based on timeline, according to the changing intensity and the nature of the communication [72]. In a single SE-related conversation thread, common patterns also include questioning an issue, discussing, and then reaching the agreement [73]. Following these prior researches, we split the percentage into three equal parts and defined each of them as a "phase": "beginning" (10%–30%), "middle"(40%–70%), and "end"(80%–100%). For example, if a conversation has 20 posts, the fifth post will be in the 25% portion of that conversation, which will be placed in the third phase.

### 3.4. Developer categorization

To answer RQ4 (Do core and non-core developers use emojis differently?), we needed to identify the status of developers in the project. Similar to prior studies, we used the number of commits contributed by individual contributors in the code base as the criterion for classifying a developer as core or non-core contributor of the project [74,75]. Open source contribution follows a power law, where 20% of contributors are responsible for 80% of the contributions [74]. Following this rule, we considered a developer as *core* if the developer is among the top 20% of developers in that project based on the number of commits authored. Otherwise, the developer is *non-core*. We found there are 91,094 core developers (22.2%) and 318,817 non-core developers (77.8%). To address the research question, we extracted developers with at least one emoji usage. We found there are 7466 core developers and 2022 non-core developers. Then, we checked whether the number of emoji usages is normally distributed in both core and non-core group using Kolmogorov Smirnov test [76] ($p$-value > 0.05). Since, the number of emoji usage is normally distributed, we conducted a two-sample t-test to check if there is a statistically significant difference on the number of emoji usages between core and non-core developers.

To further study if developers' experience level affect the intention of emoji usage, we grouped all emoji usages into six intention categories. If a developer has multiple emoji usages associated to different intentions, then the total number of emoji used by that developer will be included in each of the intention group. As the result, we found *Object Representation* with 6460 developers, *Visual Enhancement* with 2412 developers, *Sentimental Strengthening* with 768 developers, *Object Replacement* with 729 developers, *Content Organization* with 418 developers, *Sentimental Addition* with 253 developers. Similar to the aforementioned step, we checked whether the number of emoji usages is normally distributed in both core and non-core group for each intention category using Kolmogorov Smirnov test [76] ($p$-value > 0.05). Next, we conducted a two-sample t-test to check if there is a statistically significant difference on the number of emoji usages between core and non-core developers for a specific intention category. We also measured the effect size using Cohen's d [77]. We used the effsize library in R (version 0.8.1) [78].

### 3.5. Survey

To compare and validate our findings, we performed an online survey with GitHub developers. In this section, we described the design of the survey, participant selection criteria and data collection.

#### 3.5.1. Survey design

We designed an online survey to gather a deep understanding of the use of emoji in software development. We expected this survey to help us understand how, when, where, and why developers are using emojis. First, we collected demographic information to understand developers' backgrounds (e.g., current age, years of professional experience, job title, etc.) We then asked about their awareness and own habit of emoji usage on the open-source platform. To compare with the distribution outputted from our intention classifier, we included a ranking part for participants to rank the six intentions based on the usage frequency. To answer the "when" question, we asked participants to identify the positions where they use emojis in a conversation. We also asked the participants about the impact of emojis on the acceptance of pull requests, speed of getting responses, and the length of discussion. For all the above questions, we also included open-ended short answers for participants to illustrate other options or perspectives. Finally, we asked on the extant of emoji usages within a team and allow participants to leave other comments they may have. The survey instrument is available in the companion website [79].

#### 3.5.2. Participant selection

For our survey, we recruited participants who use emojis in GitHub. We identified the participants from the list of users who participated in the pull request discussions. Then, we crawled their email addresses using the GitHub API [80] based on their usernames. Overall, we identified 2,995 valid unique email addresses belongs to users who have emoji usages on the GitHub.

#### 3.5.3. Pilot survey

To help ensure the validity of the survey, we asked Computer Science professors and graduate students (Two professors and five Ph.D. students) with the experience in OSS and in survey design to review the survey. To make sure that the questions are clear and complete, we conducted several iterations of the survey and rephrase some questions according to their feedback. We also focused on the time limit to ensure that the participants could finish it under 10 min. The survey is anonymous but at the end of the survey, we gave the participants a choice to receive a summary of the study through email.

#### 3.5.4. Data collection

After sending an invitation email to 2,995 potential participants, 2,781 invites were delivered and 214 of these were not successfully delivered. We received 30 responses from 2,781 email requests during the first 10 days. Then we sent a reminder email. After the reminder, we received 20 more responses in the next 10 days. We sent out a second reminder email 10 days after the first reminder and got 10 additional responses. In total, we received 60 responses from 2,781 email requests (2.2% response rate). Our survey respondents are from 17 countries across five continents

and respondents' ages vary from 18 to 50. Also, 55% of our respondents have more than ten years of software development experience, and they are from different job roles, including software engineers, project managers, software architects, software tester, and so on. Though we only have 60 survey responses, our respondents cover a wide range of demographics.

#### 3.5.5. Data analysis

Our survey was conducted using Google Forms, and all results were automatically outputted in the histogram or pie chart form. To evaluate participants' agreement of positive effects of emoji usage in pull request acceptance, we used speed of response and conversation length. We collected the ratings provided by our respondents for each question. We converted these ratings to Likert scores from 1 (Strongly Disagree) to 5 (Strongly Agree) and computed the average Likert score. We also extracted comments and texts from the "other" fields by the survey respondents explaining the reasons behind their choices. Finally, for open-ended questions, we grouped respondents by manually detecting their agreements with the prompt. For instance, "Do you think there is a relationship between a developer's experience and emoji use? If yes, please explain"., we will categorize respondents into three groups: 1) Agree there is a relationship, 2) Do not agree there is a relationship, and 3) Not sure.

## 4. Results

Here we discuss the results of our study by placing them in the context of five research questions, which investigate the ability to predict the intention of emoji usage using machine learning techniques (RQ1), the prevalence of intentions for using an emoji (RQ2), in which phase of a conversation an emoji is used (RQ3) and who tends to use emojis (RQ4).

### 4.1. RQ1: How well can we identify the intentions of emoji usage using machine learning?

To answer this research question, we train four different machine learning algorithms: Decision Tree (DT), Multinomial Logistic Regression (MLR), Random Forest (RF), and Support Vector Machine (SVM). Among these four classifiers, Random Forest outperforms other machine learning classifiers with the highest AUC of 0.97, and highest F1-score of 0.81. The second-best performing classifier is SVM with an AUC of 0.91 and the F1-score of 0.68. Table 4 shows the performance for all classifiers in terms of precision, recall, F1-score, and AUC.

**Table 4**
Performance of the classifiers.

| | Precision | Recall | F1-score | AUC |
|---|---|---|---|---|
| RF | **0.81** | **0.81** | **0.81** | **0.97** |
| SVM | 0.73 | 0.68 | 0.68 | 0.91 |
| MLR | 0.64 | 0.63 | 0.63 | 0.87 |
| DT | 0.74 | 0.73 | 0.73 | 0.84 |

**Table 5**
Performance of random forest classifier.

| | Precision | Recall | F1 score | AUC |
|---|---|---|---|---|
| Content Organization | 0.91 | 0.98 | 0.94 | 0.99 |
| Object Replacement | 0.89 | 0.85 | 0.87 | 0.98 |
| Object Representation | 0.93 | 0.81 | 0.87 | 0.97 |
| Sentimental Addition | 0.66 | 0.71 | 0.68 | 0.93 |
| Sentimental strengthening | 0.61 | 0.68 | 0.65 | 0.93 |
| Visual Enhancement | 0.89 | 0.79 | 0.84 | 0.97 |
| AVG | 0.81 | 0.80 | 0.81 | 0.97 |

**Table 6**
Random forest classifier feature importances.

| Feature | MDI score |
|---|---|
| Position of emoji | 0.26 |
| Similarity between emoji definition and statement | 0.17 |
| Rate of neutral emoji usage | 0.12 |
| Polarity of statement | 0.12 |
| Frequency of individual emoji | 0.12 |
| Rate of negative emoji usage | 0.10 |
| Rate of positive emoji usage | 0.08 |
| Emoticons | 0.03 |

**Observation 1.** Random Forest Classifier can identify the intention of the emojis with an average AUC of 0.97.

Since the Random Forest classifier outperforms other classifiers, we use it to identify the posts' intentions with emojis from the GHTorrent dataset. In Table 5, we report the precision, recall, F1-score, and AUC of the Random Forest classifier for each of the six intention categories. On average, the Random Forest classifier has moderate precision (0.81) and high AUC (0.97). However, when looking at the performance for each category, we can see that four of them (CO, O. Repl, O. repr, and VE) have high precision. However, SA and SS have a lower precision (0.66 and 0.61). We posit that due to the reliance on the text-based sentiment analysis technique (*TextBlob*), any imprecision in the sentiment analysis technique will impact SA and SS's precision. However, to make any conclusive remarks, further investigation is required.

Our next step is to understand which of the features play the most significant role in classification. For this purpose, we use Mean Decrease in Impurity (MDI) also known as Gini Importance [81]. MDI calculates the rate of how many times a feature is included in the total number of splits across a tree model. The value of MDI ranges from 0 to 1 which greater number indicates more number of samples a feature is included for splitting. In Table 6, we report the features' importance score. The total MDI scores across all the features sum up to one, and each score suggests how useful the feature is for predicting the intention of emoji usage. The result indicates that *Position of emoji* and *Similarity between emoji definition and statement* are two of the most important features. Since *Position of emoji* considers both the emoji and the context of using the emoji, it is reasonable that this feature ends up being the most important one.

### 4.2. RQ2: What intentions do developers have while using emojis during conversations?

Our next research question is about understanding the prevalence of intentions for using an emoji. To answer this question, we investigate the intentions of emoji usage (labeled by the classifier mentioned in

**Table 7**
Emoji usage intention distribution in percentage.

| | Pull request | Issue | Commit | GitHub discussion |
|---|---|---|---|---|
| Content Organization | 0.40% | 2.01% | 17.13% | 0.00% |
| Object Replacement | 24.50% | 23.16% | 30.63% | 34.26% |
| Object Representation | 29.57% | 24.54% | 38.04% | 17.12% |
| Sentimental Addition | 0.42% | 0.41% | 1.11% | 24.07% |
| Sentimental Strengthening | 3.24% | 2.76% | 6.09% | 20.37% |
| Visual Enhancement | 41.87% | 47.12% | 7.01% | 4.17% |

RQ1) on the dataset consisting of four types of data (30,046 pull request comments, 5,908 issue comments, 3,778 commit messages, and 248 discussion comments, presented in Table 1). Due to the different usages of those communication mediums on GitHub, we expected to see a varied distribution. The labeled result of the intention distribution (in percentage) for each type of data is shown in Table 7. Based on the result from chi-square goodness test, pull request comments and issue comments are not significantly different on the intention distribution ($p$-value = 0.61) while all the other types of dataset are significantly different from each other ($p$-value $< 2.2e-16$). Since we conducted multiple statistical tests, we did the Bonferroni correction [82] and the previous chi-square goodness test resulting p-values were all smaller than the adjusted significance level of 0.008.

We found that developers use emojis in pull requests and issue comments mostly for *Visual Enhancement* purposes, 41.87% and 47.12% respectively. In commit messages, emojis are more likely to be used for *Object Representation*, *Object Replacement*, and *Content Organization*. Also, we found that developers tend to use emojis for all sorts of intentions, except for using *Content Organization* in the GitHub Discussion.

**Observation 2.** Developers' intention of using emojis differs based on which channel emojis are used on, while pull request and issue shares the same distribution due to their similar functionalities on GitHub.

*Survey:* To validate our findings through mining, we ask the survey respondents to rank each intention on a scale of 1 (least used) to 5 (most used). We calculated the mean rank responded by participants for each intention and found that *Sentiment Addition*, *Sentiment Strengthening*, and *Visual Enhancement* are the top three intentions for emoji usage (shown in Fig. 2). The result indicates that though *Visual Enhancement* is the most prevalent intention of using emoji according to our mining result, human participants report that *Sentiment Addition* is the most frequent intention of their emoji usage. We posit that participants have a biased perception regarding using emojis mostly for expressing sentiment; however, our mining result draws a different picture where emojis are used for varying intentions.

### 4.3. RQ3: When do developers use emojis the most during conversations?

To answer this question, we took a closer look at the conversation during the development process by analyzing the posts in issues, pull requests, and the GitHub discussion. We do not consider commits since there are no back and forth conversation in commits.

We only select posts within the conversations with at least ten comments. We decide to use the threshold value of ten to ensure that the comments are evenly distributed among all phases. In total, we collect 23,880 posts from pull requests and issues (12,074 posts filtered out), and 55 posts from GitHub Discussions (193 posts filtered out) from our scrapped dataset. Fig. 3 shows the distribution of emoji usages in different phases of a conversation, and we can see that developers use emojis in all phases of the conversation. We also check in which phase of the conversation developers use emoji for the first time. Similar to the previous finding, developers tend to start using emojis in all phases of the conversation.

**Observation 3.** Emojis tend to be used during every phase in a conversation on GitHub.

Then, we investigate if the distribution of intention is different depending on the phase of using emojis. Fig. 4 (Top figure) shows that *Sentimental Strengthening* tend to have a boost at the end of the conversation compared to *Content Organization* and *Sentimental Addition*. Among the three most frequently used intentions (Bottom figure), *Visual Enhancement* is always the highest except towards the end of the conversation, with a decreasing trend, while *Object Replacement* and *Object Representation* have an increasing trend towards the end of the conversation.

*Survey:* 63.6% of our survey respondents mentioned that they use emojis in any stage of the conversation they feel appropriate and related to the context of the message. When there is less emotion expressed in technical conversations, often refers to short and quick agreement messages, they tend to use emojis to add some sentiment and friendliness to the sentence. Similar to the phase distributions shown in Fig. 3, the rest of the participants mostly agree with the position of the emoji usage towards the end of the conversation as a nice way to end a conversation.

### 4.4. RQ4: Do core and non-core developers use emojis differently?

To investigate this question, we first categorized the developers into core and non-core groups (See Section 3.4 for details). We also collected the data about how frequently users use emojis in pull requests, issues, and commits. We excluded the GitHub Discussion dataset since they are individually scrapped with different time range, developers, and projects compared to the GHTorrent dataset.

We performed a two-sample t-test to see if there is a statistically significant difference on the number of emoji usages between core and non-core developers. For the core developer group, we found a mean of 1.95 emoji usages for each project they belongs to, and a mean of 1.48 for the non-core developer group. With $p$-value (0.0008) $< 0.05$ and a negligible effect size (0.08) (Cohen's d [77]), it was statistically significant that there is a small difference on the number of emoji usages between developers with different experience levels.

Then, we studied if the developer's experience level will affect the intention for their emoji usage. We grouped all emoji usages into six intention categories. We performed the two sample t-test for each intention category between core and non-core developer groups and found that there is only one statistically significant difference with a negligible effect size (0.13) (Cohen's d [77]) on the number of emoji usages between core and non-core developers for *Object Representation*, with a mean of 1.66 emoji usages for core developers and a mean of 1.27 for non-core developers. Since we conducted multiple statistical tests, we did the Bonferroni correction [82] and the resulting p-values were smaller than the adjusted significance level. Referring to Fig. 5, we found that the result distributions do not vary between the core and non-Core groups.

**Observation 4.** Developers' role as a core or non-core contributor of a project does not impact the number of emoji usages and the intention of emoji usage.

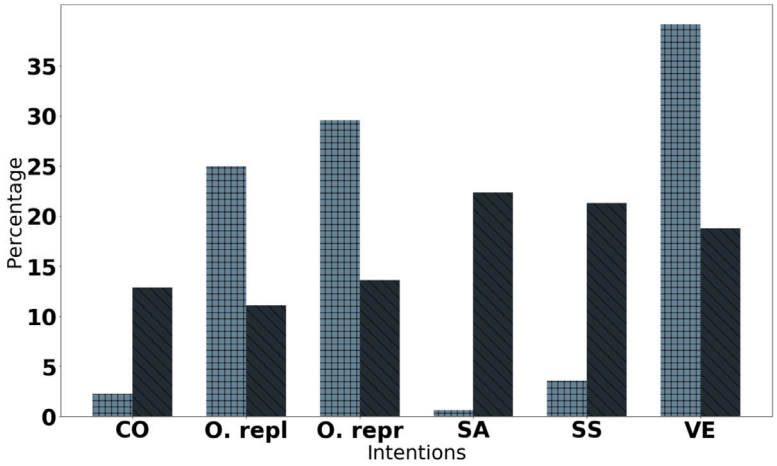



**Fig. 2.** Comparison of emoji usage intentions distribution between mining and survey results, (1) : mining distribution, (2) : survey distribution; **CO**: Content Organization, **O. Repl**: Object Replacement, **O. Repr**: Object Representation, **SA**: Sentimental Addition, **SS**: Sentimental strengthening, **VE**: Visual Enhancement.

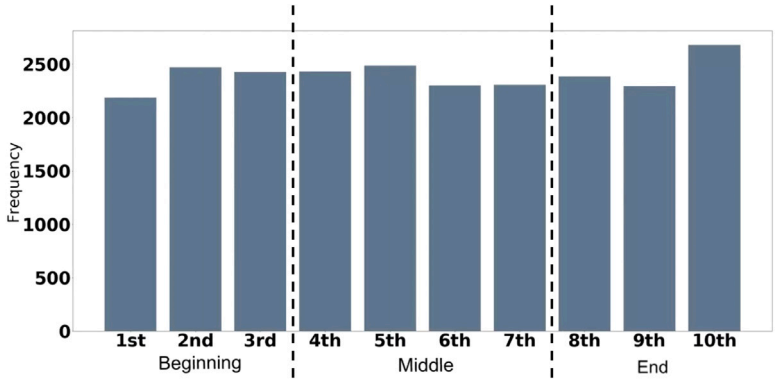



**Fig. 3.** Frequency of Emoji usages in each phase of conversations.

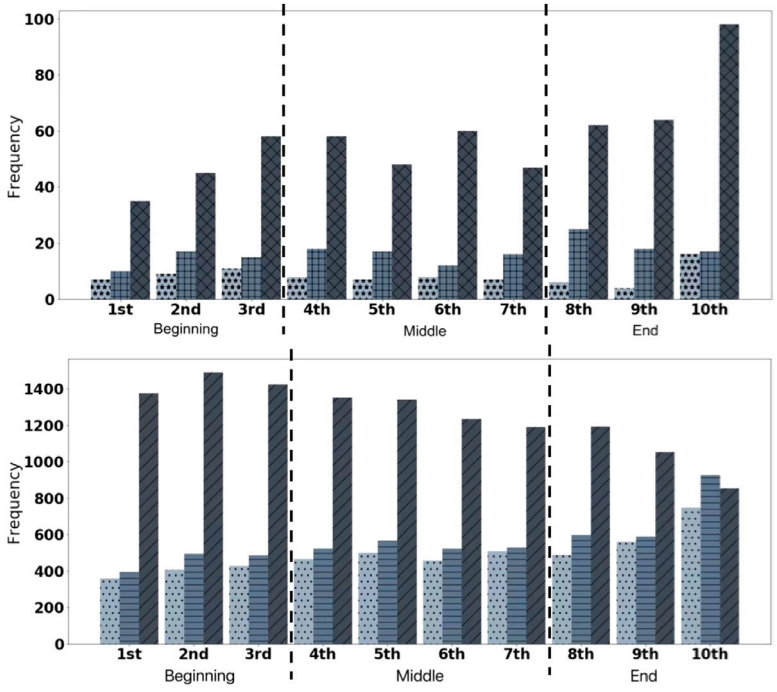



**Fig. 4.** Phase percentage of Emoji usages grouped by intentions from 23,880 pull request and issue posts, and 55 GitHub Discussion posts. Top: : SA, : CO, : SS; Bottom: : O. repl, : O.repr, : VE..

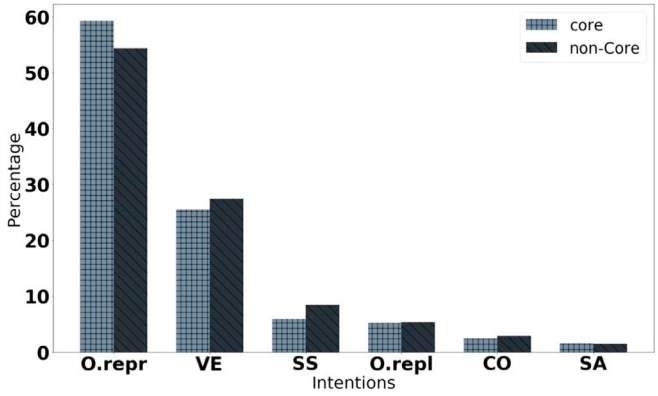



**Fig. 5.** Intention distribution for Core and non-Core developers.

*Survey:* In our survey, we asked the participants if there is any relationship between a developer's experience and emoji usage. 28.3% of the survey respondents believe there is some relationship between developer experience and emoji usages and mentioned that experienced developers are less likely to use emojis. On the other hand, 60% of the survey respondents mentioned that there is no relationship between developer status and emoji usages, and 11.7% remain neutral on this.

## 5. Discussion and implications

In this section, we discuss the results presented in the previous section and present practical implications of our study for researchers and tool builders.

### 5.1. Implication for tool builders

The intention distributions show that the intention of using emojis changes through different communication mediums. People will want to contribute to a project by either pointing out a problem using issues or simply suggesting some improvements via pull requests; therefore, *Visual Enhancement* is more likely used to attract people to look into their ideas. As the process moves forward, when developers discuss with each other, all the intentions except *Content Organization* are used more uniformly. Like traditional social media discussions, people are less likely to care about organizing content in these quick-flow discussions. Once the discussions are done, and their ideas are concrete, developers are about to submit their codes accompanying the commit messages; the intention of using emojis then switches to object-related ones and *Content Organization* especially, as sentimental related intentions are no longer useful for summarizing the code functionalities.

Based on our findings that the intentions for using emojis vary throughout conversations, tool builders can use such patterns to build an emoji recommendation system depending on the development phase and context. For example, when writing commit messages, the recommendation system can recommend emojis like "➡️" and "✅" for *Content Organization*; when users are writing bug fixes through pull requests, this system can recommend emojis like "🧯" and "⚠️" for *Visual Enhancement* to emphasize critical usages and functions get fixed. This will not only enrich developers' communication environment but also facilitate the whole development process. Tool builders could also use the result of our classifier as a feature for other sentimental analysis tasks.

### 5.2. Implication for researchers

While we found that some guidelines for using emojis in commit messages have been proposed [14–16], mostly for non-emotional intentions, guidelines for using emojis in all conversations on GitHub should also be proposed similar to the code of conduct. Based on the intention of using emojis, especially for sentimental intentions, emojis can have double meanings, for example, when expressing irony, which may hurt individuals [83]. Therefore, future researchers could also focus on predicting the intention of improper use of emojis and conduct guidelines for maintaining a healthy development atmosphere.

The intention of emoji usage also changes over time. For example, "😂" used to be the most popular emoji used for expressing the sentiment of joy; however, as new emojis and new interpretations emerged, "🤣" and "😭" started to be prevalent among younger people as a way to express an extreme and dramatic sentiment of joy and laughing.[5] As the culture of emoji usage continually updates over time, the intention of emoji usage on GitHub will also be changed. Making it necessary for project leads to track the changing habits of developers' emoji usage in order to identify the mental state of a project and, in case of improper emoji use, take necessary steps to moderate it.

The disagreement between survey results compared to our mining results could have happened due to the difference between developers' perceptions and reality, which is not uncommon. There are examples where long-held beliefs proved to be incorrect when actual evidence was collected through empirical analysis [84]; the low precision of SA and SS from Section 4.1 could also help explain this phenomenon.

## 6. Threats to validity

While we structured our study so to avoid introducing bias and have worked to eliminate the possible effects of random noise, it is possible that our mitigation strategies may not have been effective. This section reviews the threats to validity to our study.

### 6.1. Internal validity

There is a possibility that there are faults in the Python code that we implemented to perform the study. We address this threat by extensively testing our implementation.

It is possible that the sentiment analysis library (*TextBlob*) we used for "Polarity of statement" feature computation could not address all the scores properly. Though it is not designed specifically for SE conversational tasks, we believe *TextBlob* still fits our domain with four different data types as a general-scale sentiment analysis tool.

### 6.2. External validity

The dataset used for the study contains conversations from a single source—GitHub. Since we pick all conversations from the GitHub, our findings may be limited to open source projects on GitHub. However, we believe that a large number of extracted conversations from a large number of projects sampled more than adequately address this concern.

[5] https://www.cnn.com/2021/02/14/tech/crying-laughing-emoji-gen-z/index.html

### 6.3. Construct validity

For the survey results, it is always possible that the participants misunderstand the survey questions. To mitigate this threat, we conduct a pilot study with experts in OSS and survey design. We updated the survey questions based on the findings of these pilot studies. It is also possible that our findings through survey may not be generalizable due to the low response rate of participants and the perceptual inconsistency [85] even after our best effort. Since the survey is conducted to validate our findings identified through mining and survey respondents cover a wide range of demographics, we believe that this does not negatively impact our findings.

We categorized the developers into core and non-core groups using a threshold of the number of commits in the code base for each developer. Some developers could have been categorized as non-core according to our criteria though they were core developers who focus on large contributions or the architecture (high-value contributions) rather than frequent contributions of commits.

For communication phase identification, we defined three phases (beginning, middle, end) in different percentile ranges. Though the number of phases could be defined in various ways, we followed the procedure in prior research [72,73] and assume that our observation is valid.

The manual analysis applied throughout the study could have introduced unintentional bias. We labeled our training and testing data manually, which could have introduced bias or mistakes due to the lack of domain expertise. To address this concern, two researchers individually labeled a significant portion of the data. Because of the high inter-rater reliability that resulted, we assume that the risk of individual bias is minimized.

We conducted multiple statistical tests to answer the research questions which increases the chance of Type I error. To minimize the risk of Type I error, we did the Bonferroni correction [82].

### 6.4. Conclusion validity

Regarding conclusion validity, though we do not found a statistically significant relationship between developers' role and the number of emoji usages, as well as the intention of emoji usage, we do discover a small difference between two developer groups. It is possible that different set of sample might result in a slightly different significant level, and could have an opposite relationship.

## 7. Conclusions and future work

In this exploratory study, we set out to study emoji usage in software development communication. As a step towards that goal, we developed a machine learning classifier, the first of its kind, for predicting the intentions of emoji usage in software development communication.

After evaluating four different classification techniques from different families, we identified "Random Forest" as the best model, with an AUC of 0.97. Through a large-scale empirical analysis, we found that developers, irrespective of their experience level, use emojis in every step of a conversation. Our results also highlight a disconnect between what developers believed about the intention of emoji usage vs. what intentions the automated technique identified, adding to the increasing number of examples showing how long-held beliefs proved to be incorrect on investigation [84,86].

Our study opens a new avenue in Software Engineering research related to automatically identifying the intention of the emoji use that can help improve the communication efficiency and help project maintainers monitor and ensure the quality of communication. Another thread of future research could look into what intentions of emoji usage or what kind of emojis are more likely to attract users and how that is associated with emoji usage diffusion [87] in different levels (threads, projects, etc.). We also provide actionable implications for researchers, tool builders, and practitioners to harness the results of our study.

Since the intention of emoji usage can still be affected by various factors, like the background of the project, personality of developers and so on, future works could focus on adding more features to the classifier and training other classifiers to support the prediction of intentions in a longer context or even a paragraph. Another direction is to add other platforms in the study where software engineering related conversations take place such as StackOverflow, Gitter and GeekforGeeks to understand how the intention of emoji usage is different on these platforms compared to Github. The replication package link of this study is provided here [79].

## CRediT authorship contribution statement

**Shiyue Rong:** Data curation, Writing – original draft. **Weisheng Wang:** Data curation, Writing – original draft. **Umme Ayda Mannan:** Writing – original draft. **Eduardo Santana de Almeida:** Writing – review & editing. **Shurui Zhou:** Writing – review & editing. **Iftekhar Ahmed:** Project administration, Supervision, Writing – review & editing.

## Declaration of competing interest

No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.infsof.2022.106912.

## Acknowledgments

We thank the anonymous reviews for their thoughtful comments and suggestions. Professor Eduardo Almeida was partially funded by INES 2.0, FACEPE grant APQ-0399-1.03/17, and CNPq, Brazil grant 465614/2014-0.

## References

[1] H. Alshenqeeti, Are emojis creating a new or old visual language for new generations? A socio-semiotic study, Adv. Lang. Lit. Stud. 7 (6) (2016).
[2] Emoji | definition of emoji by oxford dictionary on lexico.com also meaning of emoji, 2020, https://www.lexico.com/en/definition/emoji (Accessed on 08 May 2020).
[3] Z. Chen, X. Lu, W. Ai, H. Li, Q. Mei, X. Liu, Through a gender lens: learning usage patterns of emojis from large-scale android users, in: Proceedings of the 2018 World Wide Web Conference, 2018, pp. 763–772.
[4] X. Lu, W. Ai, X. Liu, Q. Li, N. Wang, G. Huang, Q. Mei, Learning from the ubiquitous language: an empirical analysis of emoji usage of smartphone users, in: Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing, 2016, pp. 770–780.
[5] F. Al Rashdi, Forms and functions of emojis in WhatsApp interaction among Omanis, (Ph.D. thesis), Georgetown University, 2015.
[6] P.K. Novak, J. Smailović, B. Sluban, I. Mozetič, Sentiment of emojis, PLoS One 10 (12) (2015).
[7] Emojit statisics [updated july 2020], 2020, https://emojipedia.org/stats/ (Accessed on 21 Augest 2020).
[8] (20+) Messenger - posts | facebook, 2021, https://www.facebook.com/messenger/posts/there-are-over-5-billion-emojis-sent-on-messenger-every-day-to-find-the-perfect-/1422879197831756/ (Accessed on 20 Feb 2021).
[9] Social media emojis & why you should use them | agorapulse, 2021, https://www.agorapulse.com/blog/should-your-agency-use-emojis-in-social-media-really/ (Accessed on 20 Feb 2021).
[10] Emojicode, 2020, https://www.emojicode.org/ (Accessed on 21 Aug 2020).
[11] Emoji . PyPI, 2020, https://www.pypi.org/project/emoji/ (Accessed on 21 Aug 2020).
[12] Node-emoji - npm, 2020, https://www.npmjs.com/package/node-emoji (Accessed on 21 Aug 2020).
[13] X. Lu, Y. Cao, Z. Chen, X. Liu, A first look at emoji usage on GitHub: an empirical study, 2018, arXiv preprint arXiv:1812.04863.
[14] Git commit message Emoji, 2021, https://gist.github.com/parmentf/035de27d6ed1dce0b36a (Accessed on 09 Feb 2021).

[15] Gitmoji | an emoji guide for your commit messages, 2021, https://gitmoji.dev/ (Accessed on 09 Feb 2021).

[16] Dannyfritz/commit-message-emoji: Every commit is important. So let's celebrate each and every commit with a corresponding emoji!, 2021, https://github.com/dannyfritz/commit-message-emoji (Accessed on 09 Feb 2021).

[17] Z. Chen, Y. Cao, X. Lu, Q. Mei, X. Liu, SEntiMoji: an emoji-powered learning approach for sentiment analysis in software engineering, in: Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, 2019, pp. 841–852.

[18] Z. Chen, Y. Cao, H. Yao, X. Lu, X. Peng, H. Mei, X. Liu, Emoji-powered sentiment and emotion detection from software developers' communication data, ACM Trans. Softw. Eng. Methodol. (TOSEM) 30 (2) (2021) 1–48.

[19] W. Brants, B. Sharif, A. Serebrenik, Assessing the meaning of emojis for emotional awareness-a pilot study, in: Companion Proceedings of the 2019 World Wide Web Conference, 2019, pp. 419–423.

[20] C. Cheng, B. Li, Z.-Y. Li, Y.-Q. Zhao, F.-L. Liao, Developer role evolution in open source software ecosystem: An explanatory study on GNOME, J. Comput. Sci. Tech. 32 (2) (2017) 396–414.

[21] N.C. Schaeffer, Conversation with a purpose—or conversation? Interaction in the standardized interview, Meas. Errors Surv. (2004) 365–391.

[22] N. Clarke, Projects are emotional: How project managers' emotional awareness can influence decisions and behaviours in projects, Int. J. Manag. Proj. Bus. (2010).

[23] G. Gousios, D. Spinellis, Ghtorrent: GitHub's data from a firehose, in: 2012 9th IEEE Working Conference on Mining Software Repositories, MSR, IEEE, 2012, pp. 12–21.

[24] M. Claes, M. Mäntylä, U. Farooq, On the use of emoticons in open source software development, in: Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, 2018, pp. 1–4.

[25] About team discussions - GitHub docs, 2020, https://docs.github.com/en/github/building-a-strong-community/about-team-discussions (Accessed on 05 Aug 2020).

[26] C.C.A. Blaz, K. Becker, Sentiment analysis in tickets for IT support, in: Proceedings of the 13th International Conference on Mining Software Repositories, 2016, pp. 235–246.

[27] D. Gachechiladze, F. Lanubile, N. Novielli, A. Serebrenik, Anger and its direction in collaborative software development, in: 2017 IEEE/ACM 39th International Conference on Software Engineering: New Ideas and Emerging Technologies Results Track, ICSE-NIER, IEEE, 2017, pp. 11–14.

[28] E. Guzman, B. Bruegge, Towards emotional awareness in software development teams, in: Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering, 2013, pp. 671–674.

[29] M. Mäntylä, B. Adams, G. Destefanis, D. Graziotin, M. Ortu, Mining valence, arousal, and dominance: possibilities for detecting burnout and productivity? in: Proceedings of the 13th International Conference on Mining Software Repositories, 2016, pp. 247–258.

[30] M. Ortu, B. Adams, G. Destefanis, P. Tourani, M. Marchesi, R. Tonelli, Are bullies more productive? Empirical study of affectiveness vs. issue fixing time, in: 2015 IEEE/ACM 12th Working Conference on Mining Software Repositories, IEEE, 2015, pp. 303–313.

[31] S. Panichella, A. Di Sorbo, E. Guzman, C.A. Visaggio, G. Canfora, H.C. Gall, How can i improve my app? classifying user reviews for software maintenance and evolution, in: 2015 IEEE International Conference on Software Maintenance and Evolution, ICSME, IEEE, 2015, pp. 281–290.

[32] R. Souza, B. Silva, Sentiment analysis of travis ci builds, in: 2017 IEEE/ACM 14th International Conference on Mining Software Repositories, MSR, IEEE, 2017, pp. 459–462.

[33] M.R. Wrobel, Emotions in the software development process, in: 2013 6th International Conference on Human System Interactions, HSI, IEEE, 2013, pp. 518–523.

[34] M.R. Wrobel, Towards the participant observation of emotions in software development teams, in: 2016 Federated Conference on Computer Science and Information Systems, FedCSIS, IEEE, 2016, pp. 1545–1548.

[35] M.R. Islam, M.F. Zibran, Leveraging automated sentiment analysis in software engineering, in: 2017 IEEE/ACM 14th International Conference on Mining Software Repositories, MSR, IEEE, 2017, pp. 203–214.

[36] N. Novielli, D. Girardi, F. Lanubile, A benchmark study on sentiment analysis for software engineering research, in: 2018 IEEE/ACM 15th International Conference on Mining Software Repositories, MSR, IEEE, 2018, pp. 364–375.

[37] B. Lin, F. Zampetti, G. Bavota, M. Di Penta, M. Lanza, R. Oliveto, Sentiment analysis for software engineering: How far can we go? in: Proceedings of the 40th International Conference on Software Engineering, 2018, pp. 94–104.

[38] R. Jongeling, S. Datta, A. Serebrenik, Choosing your weapons: On sentiment analysis tools for software engineering research, in: 2015 IEEE International Conference on Software Maintenance and Evolution, ICSME, IEEE, 2015, pp. 531–535.

[39] T. Ahmed, A. Bosu, A. Iqbal, S. Rahimi, SentiCR: a customized sentiment analysis tool for code review interactions, in: 2017 32nd IEEE/ACM International Conference on Automated Software Engineering, ASE, IEEE, 2017, pp. 106–111.

[40] M.R. Islam, M.F. Zibran, SentiStrength-SE: Exploiting domain specificity for improved sentiment analysis in software engineering text, J. Syst. Softw. 145 (2018) 125–146.

[41] F. Calefato, F. Lanubile, F. Maiorano, N. Novielli, Sentiment polarity detection for software development, Empir. Softw. Eng. 23 (3) (2018) 1352–1382.

[42] Scrapy | a fast and powerful scraping and web crawling framework, 2020, https://scrapy.org/ (Accessed on 05 Aug 2020).

[43] emoji, https://pypi.org/project/emoji/ (Accessed: 24 Aug 2020).

[44] SEntiMoji/SEntiMoji: data, code, pre-trained models and experiment results for "sEntiMoji: An emoji-powered learning approach for sentiment analysis in software engineering", 2020, https://github.com/SEntiMoji/SEntiMoji (Accessed on 05 Aug 2020).

[45] J.L. Campbell, C. Quincy, J. Osserman, O.K. Pedersen, Coding in-depth semistructured interviews: Problems of unitization and intercoder reliability and agreement, Sociol. Methods Res. 42 (3) (2013) 294–320.

[46] J.L. Fleiss, Measuring nominal scale agreement among many raters, Psychol. Bull. 76 (5) (1971) 378.

[47] Fleiss' kappa - wikipedia, 2021, https://en.wikipedia.org/wiki/Fleiss%27_kappa (Accessed on 08 Jan 2021).

[48] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, J. Artificial Intelligence Res. 16 (2002) 321–357.

[49] Full Emoji list, v13.1, 2021, https://unicode.org/emoji/charts/full-emoji-list.html (Accessed on 22 Feb 2021).

[50] S. Bird, E. Klein, E. Loper, Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit, O'Reilly Media, Inc., 2009.

[51] Textblob . PyPI, 2020, https://pypi.org/project/textblob/ (Accessed on 05 Aug 2020).

[52] G. Forman, BNS feature scaling: an improved representation over tf-idf for svm text classification, in: Proceedings of the 17th ACM Conference on Information and Knowledge Management, 2008, pp. 263–270.

[53] Decision trees, 2019, https://scikit-learn.org/stable/modules/tree.html (Accessed: 17 Aug 2019).

[54] Logistic regression, 2019, https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html (Accessed: 17 Aug 2019).

[55] Randomforestclassifier, 2019, https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html (Accessed: 17 Aug 2019).

[56] Support vector machine, 2019, https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html (Accessed: 17 Aug 2019).

[57] A.S. Zharmagambetov, A.A. Pak, Sentiment analysis of a document using deep learning approach and decision trees, in: 2015 Twelve International Conference on Electronics Computer and Computation, ICECCO, IEEE, 2015, pp. 1–4.

[58] A. Prabhat, V. Khullar, Sentiment classification on big data using naïve Bayes and logistic regression, in: 2017 International Conference on Computer Communication and Informatics, ICCCI, IEEE, 2017, pp. 1–5.

[59] K. Sridharan, G. Komarasamy, Sentiment classification using harmony random forest and harmony gradient boosting machine, Soft Comput. 24 (10) (2020) 7451–7458.

[60] S. Naz, A. Sharan, N. Malik, Sentiment classification on twitter data using support vector machine, in: 2018 IEEE/WIC/ACM International Conference on Web Intelligence, WI, IEEE, 2018, pp. 676–679.

[61] D. Coppersmith, S.J. Hong, J.R. Hosking, Partitioning nominal attributes in decision trees, Data Min. Knowl. Discov. 3 (2) (1999) 197–217.

[62] D.C. Liu, J. Nocedal, On the limited memory BFGS method for large scale optimization, Math. Program. 45 (1) (1989) 503–528.

[63] scikit-learn, 2019, http://scikit-learn.org/stable/ (Accessed: 1 Aug 2019).

[64] G. James, D. Witten, T. Hastie, R. Tibshirani, An introduction to statistical learning, Vol. 112, Springer, 2013.

[65] M. Kuhn, K. Johnson, Applied Predictive Modeling, Vol. 26, Springer, 2013.

[66] J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization, J. Mach. Learn. Res. 13 (Feb) (2012) 281–305.

[67] A. Bernstein, J. Ekanayake, M. Pinzger, Improving defect prediction using temporal features and non linear models, in: Ninth International Workshop on Principles of Software Evolution: in Conjunction with the 6th ESEC/FSE Joint Meeting, ACM, 2007, pp. 11–18.

[68] D. Di Nucci, F. Palomba, G. De Rosa, G. Bavota, R. Oliveto, A. De Lucia, A developer centered bug prediction model, IEEE Trans. Softw. Eng. 44 (1) (2018) 5–24.

[69] E. Giger, M. D'Ambros, M. Pinzger, H.C. Gall, Method-level bug prediction, in: Proceedings of the ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, ACM, 2012, pp. 171–180.

[70] E. Giger, M. Pinzger, H.C. Gall, Comparing fine-grained source code changes and code churn for bug prediction, in: Proceedings of the 8th Working Conference on Mining Software Repositories, ACM, 2011, pp. 83–92.

[71] F. Zhang, A. Mockus, I. Keivanloo, Y. Zou, Towards building a universal defect prediction model, in: Proceedings of the 11th Working Conference on Mining Software Repositories, ACM, 2014, pp. 182–191.

[72] A. Sfard, C. Kieran, Cognition as communication: Rethinking learning-by-talking through multi-faceted analysis of students' mathematical interactions, Mind Cult. Activity 8 (1) (2001) 42–76.

[73] G. Bougie, J. Starke, M.-A. Storey, D.M. German, Towards understanding twitter use in software engineering: preliminary findings, ongoing challenges and future questions, in: Proceedings of the 2nd International Workshop on Web 2.0 for Software Engineering, 2011, pp. 31–36.

[74] A. Mockus, R.T. Fielding, J.D. Herbsleb, Two case studies of open source software development: Apache and mozilla, ACM Trans. Softw. Eng. Methodol. (TOSEM) 11 (3) (2002) 309–346.

[75] I. Ahmed, U.A. Mannan, R. Gopinath, C. Jensen, An empirical study of design degradation: How software projects get worse over time, in: Empirical Software Engineering and Measurement (ESEM), 2015 ACM/IEEE International Symposium on, IEEE, 2015, pp. 1–10.

[76] F.J. Massey Jr., The Kolmogorov-Smirnov test for goodness of fit, J. Amer. Statist. Assoc. 46 (253) (1951) 68–78.

[77] Cohen's D: Definition, examples, formulas - statistics how to, 2020, https://www.statisticshowto.com/cohens-d/ (Accessed on 18 Aug 2020).

[78] Cohen.d function - rdocumentation, 2022, https://www.rdocumentation.org/packages/effsize/versions/0.8.1/topics/cohen.d (Accessed on 04 Feb 2022).

[79] GitHub - Emojiintentionclassifier/emoji-intention-classifier, 2021, https://github.com/EmojiIntentionClassifier/Emoji-Intention-Classifier (Accessed on 21 Feb 2021).

[80] Activity - GitHub docs, 2020, https://docs.github.com/en/rest/reference/activity#list-public-events-for-a-user (Accessed on 17 Aug 2020).

[81] B.H. Menze, B.M. Kelm, R. Masuch, U. Himmelreich, P. Bachert, W. Petrich, F.A. Hamprecht, A comparison of random forest and its gini importance with standard chemometric methods for the feature selection and classification of spectral data, BMC Bioinformatics 10 (1) (2009) 213.

[82] Y. Benjamini, D. Yekutieli, The control of the false discovery rate in multiple testing under dependency, Ann. Statist. (2001) 1165–1188.

[83] T. Hu, H. Guo, H. Sun, T.-v.T. Nguyen, J. Luo, Spice up your chat: The intentions and sentiment effects of using emoji, 2017, arXiv preprint arXiv:1703.02860.

[84] P. Devanbu, T. Zimmermann, C. Bird, Belief & evidence in empirical software engineering, in: 2016 IEEE/ACM 38th International Conference on Software Engineering, ICSE, IEEE, 2016, pp. 108–119.

[85] K.L. Ailawadi, R. Dant, D. Grewal, Perceptual and objective performance measures: an empirical analysis of the difference and its impact, Tuck School of Business at Dartmouth Administration, Research Paper Series (2003) Forthcoming.

[86] T. Menzies, W. Nichols, F. Shull, L. Layman, Are delayed issues harder to resolve? Revisiting cost-to-fix of defects throughout the lifecycle, Empir. Softw. Eng. 22 (4) (2017) 1903–1935.

[87] H. Lamba, A. Trockman, D. Armanios, C. Kästner, H. Miller, B. Vasilescu, Heard it through the Gitvine: an empirical study of tool diffusion across the npm ecosystem, in: Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, 2020, pp. 505–517.