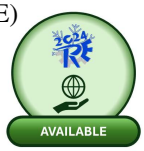
2024 IEEE 32nd International Requirements Engineering Conference (RE)



Uncovering Patterns in Users' Ethical Concerns about Software

Özge Karaçam
Vrije Universiteit Amsterdam
ozgekaracam2@gmail.com

Tom P. Humbert
Vrije Universiteit Amsterdam
t.humbert@vu.nl

Emitzá Guzmán
Vrije Universiteit Amsterdam
e.guzmanortega@vu.nl

2024 IEEE 32nd International Requirements Engineering Conference (RE) | 979-8-3503-9511-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/RE59067.2024.00055

***Abstract*—Ethical concerns about software applications, e.g., worries about privacy breaches, user manipulation, and discrimination, have gained prominence recently. Research shows that users voice these concerns in app reviews and that they can be detected using machine learning and deep learning techniques. These techniques usually operate as black-boxes, making it difficult to understand the context of users' ethical concerns. We address this issue by presenting a transparent approach that uses pattern mining and graph theory to yield additional context to the ethical concern classifications made by machine learning algorithms. We compare a simple frequent pattern mining and a high-utility mining algorithm and assess the resulting rules through commonly used metrics. Finally, we visualize and interpret preliminary results in an interactive graph. We mined 3,101 reviews of ten popular apps mentioning diverse ethical concerns and present the results for two apps in detail. Our results show that pattern mining algorithms and graph visualizations are promising directions for detecting contextual information of ethical concerns about software. This work is a step toward ensuring that ethical concerns are methodically thought through and integrated into the software development life cycle.**

I. Introduction

Software has become an integral part of our daily lives. New uses for software are continually discovered. Nevertheless, as software usage grows, so does the amount of *ethical concerns* about software. We define ethical concerns about software as end-users' worries about wrongdoing by or through software. This wrongdoing can affect individuals or society as a whole. Studies show that end-users express a wide range of ethical concerns about software applications, including but not limited to data privacy, accountability, and discrimination [1], [2], [3], [4], [5], [6]. Ethical concerns of users may or may not be founded by accurate knowledge, yet none should be discarded from analysis preemptively. Any concern may add insight to the complex relationship people develop with software.

Despite the prevalence of ethical concerns about software, there is still no systematic approach for considering unaddressed ethical concerns during software development. Previous research [4], [6] shows that these ethical concerns are often reported in user feedback, such as app store reviews and social media posts. Classification techniques have been used to filter and categorize this feedback by their predominant ethical concerns, offering a general overview using predefined labels.

While these classification approaches are valuable for filtering and detecting ethical concerns, they provide little *context*. We use the term context to describe the information that surrounds ethical concerns, e.g., entities, actions, or lingo. Moreover, we argue that different ethical concerns could share a common context.

Take for example the ethically concerned TikTok users in Figure 1. As active members of society, most of us should get a vague sense of how these reviews came to be. Yet, to derive software requirements that address ethical issues, we should not be led by personal biases. Often, we cannot take what is said at face value either, nor do upvotes or star ratings always coherently reflect what is important and what is not. What we can do is uncover the hidden subtleties of speech, i.e., recurring speech patterns that encode a shared understanding. In our example (cf. Figure 1), we hypothesize that review A and B have such a shared understanding, and it is likely they point at users such as reviewer C. Despite their differences, these reviews appear to share a common context. To systematically consider ethical concerns during requirements elicitation and software evolution it is crucial to understand the nature of the underlying context.

In this work, we take an initial step in this direction. We analyze what users talk about, and how it relates to what others say when communicating ethical concerns about software. We go beyond labeling reviews using black-box classifiers and apply *pattern mining techniques* and *graph theory* to uncover the underlying context of user reviews describing ethical concerns. For this purpose, a classification label for a review is only considered to be one more contextual information. We systematically extract patterns of contextual information, i.e., frequent and interesting associations between entities, actions, or descriptions from user reviews. Afterward, we visualize our results using interactive graphs.

To our best knowledge, this study is the first to apply this set of techniques to user feedback from an ethics perspective. We believe that pattern mining and graph visualizations can uncover the nature of ethical concerns by establishing and putting them into context, improving researchers' and practitioners' understanding of ethical concerns about software. Likewise, it can aid in the discovery of emerging, unrecorded types and sub-types of ethical concerns, thereby furthering the ongoing effort of building a comprehensive ethical concerns taxonomy for software development.

This work is a step toward ensuring that ethical concerns are methodically thought through and integrated into the software development life cycle. Disregarding this responsibility may

2332-6441/24/$31.00 ©2024 IEEE
DOI 10.1109/RE59067.2024.00055

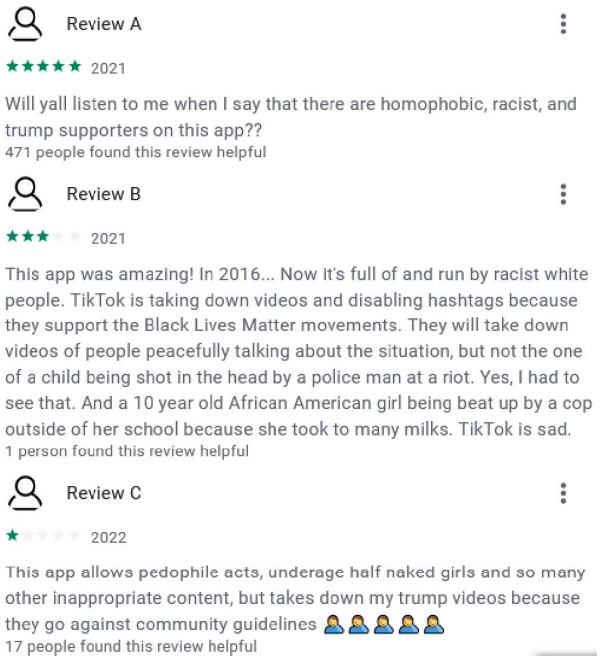


Fig. 1: Three reviews with different star ratings and varying amounts of peer support. In previous work [4], Reviews A and B are classified as a discrimination concern, and C as an inappropriate content concern. Review A denounces discriminatory Trump supporters. Review B claims the company is now run by white racists and silences those who are discriminated against. Review C asks why their Trump videos are reported and taken down, pointing out that there is inappropriate content to be removed. Despite their differences, these reviews appear to share a common context.

lead to a loss of user loyalty and an overall loss of stakeholder trust. Note that we assume the **software organizations' responsibility for their users' experience on the platform**. Software applications capitalize on user behavior and content to run their services. Due to this intertwinedness, software practitioners must acknowledge their responsibility to impose ethically sound structures that ensure, e.g., appropriate content and safe interactions through their designs.

In this study, we provide preliminary answers to the research question: *Is frequent pattern mining an effective technique for providing context about ethical concerns expressed in user reviews?* The contributions of this work are (1) an approach to identify interesting and frequent ethical concern patterns in user reviews, (2) an interactive network graph visualization illustrating patterns surrounding ethical concerns, and (3) a preliminary assessment of the effectiveness of the approach. We make our replication package available.[1]

[1]https://doi.org/10.6084/m9.figshare.25204388

II. Related Work

Previous work found that user feedback is essential for software quality and identifying areas of improvement [7]. Hoon [8], Pagano, and Maalej [9] were among the first to study user feedback from app reviews. They performed exploratory studies and found that app reviews contain valuable information for requirements elicitation and software evolution. Pagano and Maalej [9] applied frequent pattern mining on manually annotated labels describing review content to analyze its main themes and underlying patterns. While we also apply pattern mining to reviews, we focus on those mentioning ethical concerns to understand its context better.

Research on automatic techniques for processing user feedback has grown considerably in recent years. One of the most studied user feedback artifacts is app reviews; Martin et al. [10] gives an overview of work in the field. For example, previous work has proposed different approaches for classifying [11], [12], [13], summarizing [14], [15], [16], [17], [18] and prioritizing [14], [15], [16], [19], as well as for extracting software features mentioned in the feedback [20], [21] and linking it to other user feedback artifacts [22]. In our work, we use similar preprocessing steps and supervised machine learning models as those described in the aforementioned previous work [11], [12], [13].

There is still few research on ethical concerns mentioned in user feedback. Most of this work has focused on single ethical concerns. Privacy is the most studied ethical concern [1], [23], [24], [2], but other concerns, such as discrimination/inclusion [3], [25] have also been studied. Besides these studies, previous work studied multiple ethical concerns mentioned in app reviews [4] and on Reddit [6]. While the former proposed an initial taxonomy for ethical concerns and applied machine learning and deep learning techniques for its classification; the latter focused exclusively on concerns expressed by marginalized communities. In this work, we use the dataset and ethical concerns identified in previous work [4] and apply pattern mining and visualization techniques to the identified ethical concerns to obtain additional context and detect overlapping themes and possible conflicts.

III. Approach

Our approach uncovers contextual patterns in ethical concerns about software expressed in user feedback, such as app reviews. As Figure 2 shows, it consists of five steps: (1) we *preprocess* the feedback, (2) we automatically *classify* it into types of ethical concerns, (3) we apply frequent and high utility-frequent *pattern mining* on the user feedback containing ethical concerns and generate **association rules**, (4) we strategically determine the **interestingness** of association rules through thresholds set on well-established metrics. After thresholding, only **strong rules** describing contextual patterns in end-users' ethical concerns remain, (5) we translate these strong rules into a directed *network graph* to visualize patterns and interpret the context of the ethical concerns. Note that Figure 2 shows the different designations of rules (association,

interesting and strong) at each step. We detail our *user feedback dataset* and each of the five steps as follows.

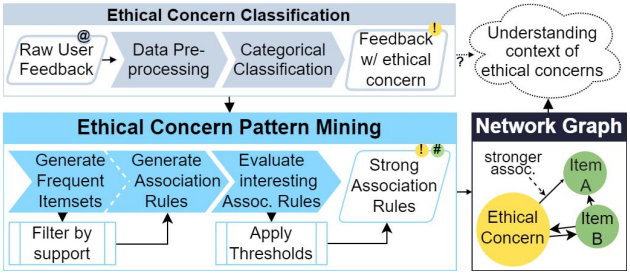


Fig. 2: Approach overview with which ethical concern patterns are extracted and visualized, after its automatic classification.

A. User Feedback Dataset

While our approach could, with minor modifications, work on diverse user feedback artifacts, we illustrate its use on user reviews that were collected and manually labeled in previous work [4]. This dataset consists of 3,101 Google Play Store reviews from ten popular apps: TikTok, Uber, Facebook, YouTube, Instagram, Linkedin, Vinted, Zoom, Alexa, and Google Home. The manually annotated labels indicate if a review contains an ethical concern and its type. Table I displays the 16 types of ethical concerns, along with their distribution percentages in the dataset. Table II shows examples of reviews and their ethical concern types. Overall, 61.9% of the reviews in the dataset included an ethical concern.

Discrimination	8.7%	Inappropriate Content	6.9%
Accountability	5.7%	Privacy	5.5%
Censorship	5.5%	Cyberbullying	4.9%
Scam	4.8%	Safety	4.6%
Spreading false information	3.6%	Addiction	3.6%
Identity Theft	3.2%	Transparency	1.5%
Harmful Advertising	1.5%	Accessibility	1.3%
Sustainability	0.4%	Content Theft	0.3%

TABLE I: Ethical concern type distribution after multi-label classification.

User Review (Platform)	**Ethical Concern**
The app **silences POC** when they try to spread awareness about certain **issues/ causes**. (TikTok)	Discrimination
Please avoid **sexual content** on app (YouTube)	Inappr. Content
I [sent them a] **report** [..] but no reply from your side it's **customer service** is **bad** [..] (TikTok)	Accountability
You tube **invades my privacy** and hijacks my phone. [..] (YouTube)	Privacy

TABLE II: Examples of ethical concerns in user reviews (shortened and anonymized) for the top four ethical concerns.

B. Preprocessing

We prepared our text by removing website links, special characters, symbols, numbers, and stopwords [26]. Using the NLTK library [27], we also lowercased and lemmatized the text and used part-of-speech tagging to prioritize verb, adjective, and noun roots.

C. Supervised Classification

We used the readily available manual labels from our dataset (see Section III-A) to train binary machine learning models and identify reviews that contain ethical concerns. Then, we used the same dataset to train additional multi-class models to identify the concrete ethical concern types, listed in Table I, that were mentioned in the review. After empirical validation of several machine learning models (Random Forest (RF), Multinomial Naive Bayes (MNB), Logistic Regression (LR), Multi-Layer Perceptron (MLP), and Support Vector Machine (SVM)), we chose the MNB and RF models for our approach as they showed the best performance. Overall, MNB achieved a Precision of 0.92, Recall of 0.87, F1-score of 0.90 for the binary classification, and RF achieved a Precision, Recall, and F1-score of 0.80, 0.82, and 0.81, respectively for the multi-class classification. We only used reviews containing ethical concerns in the following steps.

D. Pattern Mining

We compare two pattern mining algorithms to identify recurrent patterns in user feedback with ethical concerns: the classic Apriori algorithm [28] and the more recent High Utility-Frequent Itemset Mining (HU-FIM) [29].

The goal of these is to identify association rules among various items in a generalized form of 'if-then' statements, where the 'if' (antecedent) condition precedes the 'then' (consequent) [30]. These rules are derived from iteratively generated frequent itemsets, which are collections of items frequently occurring together in transaction data [31].

In our work, we treat each *user review* as a *transaction* and consider individual *words* as *items*. We grouped reviews separately for each app to understand association rules within an app-focused context. We included the identified ethical concern types as items in each review by appending the term *_concern* to the review type (e.g., *discrimination_concern* and *scam_concern*) and distinguish it from ordinary words. We removed app names from reviews, as we have individual corpora per app. Finally, each corpus was one-hot encoded. We used the same corpora in both mining algorithms.

We used the Python Mlxtend library [32] to apply the Apriori algorithm to our dataset. Apriori uses a bottom-up approach to iteratively generate frequent k-itemsets for increasing values of k, starting from $k = 1$. We recorded the support values, i.e., word frequencies, of unique words in the first iteration to establish a suitable support threshold for the next iteration. Items with very low support are not likely to contribute to interesting rules in subsequent steps [28]. The Apriori principle says that a k-itemset is considered frequent only if all of its (k-1) sub-itemsets are also frequent [31], [30]. We determined the support values per app corpus in Section IV.

HU-FIM prunes irrelevant itemsets by applying both support and utility thresholds. We used the implementation from the Python PAMI library [33]. Like Apriori, the intuition behind utility mining is rooted in market basket analysis. But, it addresses the issue of frequent patterns of cheap items, such as {milk, bread} that are not as useful to make profits as frequent patterns of expensive items. Analogously, we argue that some words and patterns thereof are less useful to establish context than others, such as {very, good}. We lemmatized words and removed stopwords before mining because of this reasoning.

Utility values present the opportunity for a fine-grained tuning of word usefulness without completely removing them from the data. For example, the stopword list we use contains the word *concerning*, which is excluded in its meaning of *regarding* but could be valuable to ethical concern context when used as a *bothering* synonym. Additional work on utility value assignment is required. For now, we assigned arbitrary utility values according to the part-of-speech of words, namely nouns:20, verbs:10, adjectives:5, and the value of 1 for anything else. This includes every word that is not a synset in Wordnet [34], [35], such as misspelled words or slang. We assigned the value of 50 to the ethical concern label item in each transaction, with the reasoning that itemsets containing it are always of high utility and should rather be pruned by support. We apply the same support thresholds that have been determined for Apriori mining. We determine the utility threshold experimentally, increasing its value in increments until the algorithm produces a comparable amount of itemsets as Apriori.

In this work, we only used 2-itemsets to identify the most fundamental frequent patterns in the reviews. We stopped Apriori at $k = 2$ and pruned $k > 2$ k-itemsets from the HU-FIM output. Finally, we generated association rules from itemsets, which we assess in terms of interestingness in the next section.

E. Strong Association Rules

We generated all association rules without setting any thresholds first to explore measures of interestingness and then used our insights to set filtering thresholds. We retain only the most interesting association rules, referred to as *strong rules*.

In this preliminary, we explore the statistical measures of antecedent and consequent support, confidence [31], lift [36], and Zhang [37] as measures of interestingness.

Support and confidence: While antecedent and consequent support values indicate the frequency of items in itemsets, confidence provides directionality, measuring the conditional probability of the consequent appearing, given the presence of the antecedent, e.g., the confidence of 0.85 for $\{girl \rightarrow naked\}$ means that 85% of concerned TikTok reviews that mentioned *girl*, also mentioned *naked*.

Lift and Zhang: In certain cases, rules may exhibit high support and confidence, yet lack practical relevance, particularly when the consequent of the rule has a greater probability of being selected independently [36]. The lift metric addresses this issue by measuring the frequency of two items occurring together relative to what would be expected if they were statistically independent. For instance, $I_1 = \{black, discrimination\_concern\}$ and $I_2 = \{black, creator\}$ are two frequent 2-itemsets with confidence value 1. However, the lift of I_1 is 3.25, whereas it is 10.7 for I_2. This means that users mentioning *black* in their review are 10.7 times more likely to point at *creator*s than expected. In terms of our interpretation, this points to a discrimination concern surrounding blackness, and more specifically, black creators. Zhang's metric is a variation of lift, given in percentage, that is mainly used to quantify the disassociation between items. It ranges from -1 to 1, where positive values signify association and negative values indicate disassociation. For instance, a Zhang value of 0.82 for $\{nudity \rightarrow inappropriate\_content\_concern\}$ means that the term *nudity* is 82% more likely to co-occur with the *inappropriate content* concern than expected. In this work, we focus only on the association and not the disassociation of rules.

F. Network Graph

In this final step, we convert strong rules into graph data and visually interpret it. We created our interactive network graphs using the d3block Python library [38]. The network graph displays the items of the association rules as nodes, while their relationships are represented as edges with arrows indicating the direction of the rule. Edge widths, based on lift values, represent association strength. Node size and opacity increase with incoming edges, and node colors create visually cohesive clusters. We currently use the integrated clustering algorithm of *d3blocks*, which chooses cluster colors randomly. We use this visualization to detect underlying themes and better understand the context of ethical concerns.

IV. Preliminary Results

Due to space limitations, we present our results for two apps: TikTok and YouTube. The TikTok corpus contains 182 transactions with 842 unique items, while the YouTube corpus contains 45 transactions with 360 unique items. Table III summarizes, per app and algorithm, the number of generated frequent itemsets, the total number of resulting association rules, the number of high confidence association rules, and the size of the final set of strong rules. Table IV further details for both apps the support and utility thresholds used in frequent itemset generation, and the confidence, lift, and Zhang thresholds applied to association rules to determine strong rules.

A. Determination of Strong Rules

We calculated each transaction item's frequency of occurrence divided by the number of total transactions, namely support, before generating frequent itemsets. In order to filter out a large number of low-frequency itemsets that may not yield significant patterns, we established support count thresholds. The thresholds are empirically set at points where the support value frequency abruptly drops (cf. Figure 3).

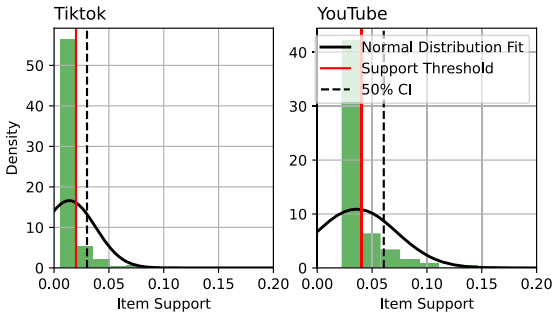


Fig. 3: Distributions of items in terms of support, i.e., how often they appear in the corpus. Since distributions differ between app datasets we empirically determine which items to keep per app. The red line represents the chosen support threshold, excluding only the first and largest bin in the histograms.

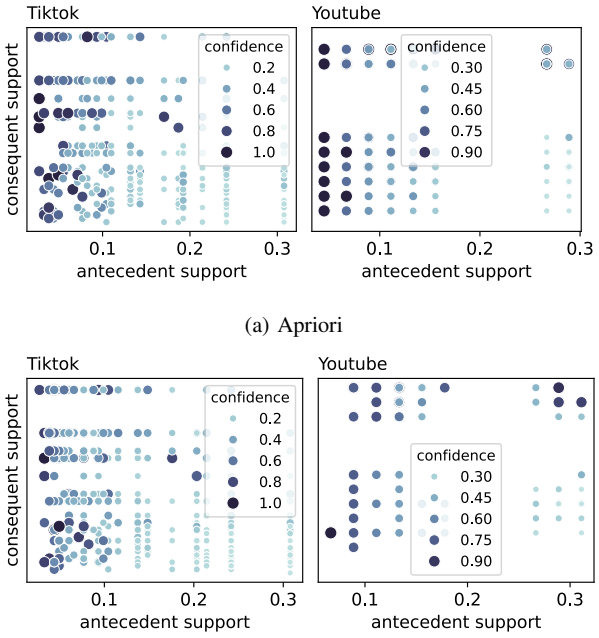
The next possible threshold coincides in both cases with the 50% confidence interval of the fitted normal distribution. We set the support thresholds to 0.02 for TikTok, meaning an item must appear in at least 4 distinct transactions. For YouTube, we chose a support of 0.04, such that an item needs to appear at least twice in the corpus. This thresholding approach ensures that only itemsets with a substantial frequency are considered, enabling the extraction of meaningful, frequent patterns from the data [28].

We generated a total of 155 itemsets for TikTok and 165 itemsets for YouTube with Apriori. HU-FIM, with the utility values given in Table IV, yielded 143 itemsets for TikTok and a considerably lower total of 62 itemsets for YouTube. However, lowering the utility threshold by one would produce an unmanageable number of 288 itemsets for YouTube. These itemsets are translated into association rules.

The association rules exhibit a symmetry, visible in Figure 4. Rules, in contrast to itemsets, have a direction; the antecedent points at the consequent. Every 2-itemset creates two distinct rules. Following the principle of having asymmetric interestingness measures in association rules [37], we used the association rules in the upper side of the symmetries in Figure 4. In this region, the rules have consequent support values equal to or higher than their antecedent support. Confidence, a measure of reliability, shows higher values when the consequent support is greater than the antecedent support. Therefore, we discarded the rules in the lower part of the symmetry. See Table III for detailed results.

Figure 5 illustrates support, confidence, and lift values for all association rules. It shows that many rules have low support but high lift, indicating that interesting rules lie at their intersection, aligning with prior research [39]. We set the confidence thresholds 0.4 for TikTok and 0.5 for YouTube (cf. Table IV) at points where the lift values began to surpass others to retain higher lift rules compared to the rest.

Figure 6 illustrates the relationship between the lift and



(a) Apriori

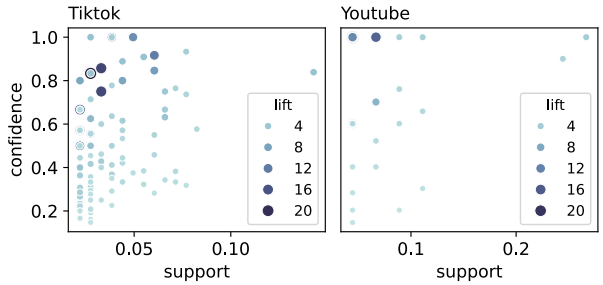
(b) HU-FIM

Fig. 4: Symmetry that association rules exhibit after they have been generated from itemsets. Each itemset results in two separate rules, of which one has higher antecedent support and the other has higher consequent support. We follow the principle of asymmetric interestingness [37] and only keep the upper triangle of the symmetry, namely rules where the consequent item has higher support. The number of rules we kept in this step can be found in the column 'Confident' of Table III. These rules also have higher confidence. By comparing the results of HU-FIM against Apriori we see that the former produces fewer rules with high confidence. With HU-FIM, there are also only half as many rules for YouTube, but they have generally higher antecedent and consequent support. We do not observe this peculiarity for TikTok.

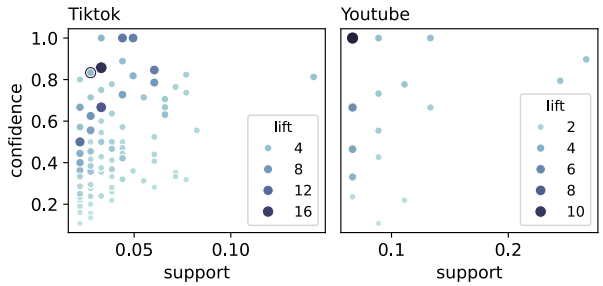
Zhang metrics. Strong association rules are characterized by higher Zhang values and significantly higher lift values. To eliminate weak association rules, we set thresholds based on where the exponential portion of the Zhang-lift curve begins. We applied a lift threshold of 2 for both apps, with Zhang thresholds of 0.5 for TikTok and 0.7 for YouTube (cf. Table IV). For the results of HU-FIM, looking at the differences between apps, the utility threshold appears to decide on the final number of strong rules. The higher utility values for concern items had no effect; we detected per app the same sets of ethical concern types as from Apriori.

B. Patterns in Network Graphs

We illustrate the strong association rules in Figure 7 as network graph visualizations, per app and mining algorithm. We discovered various patterns relating to ethical concerns in

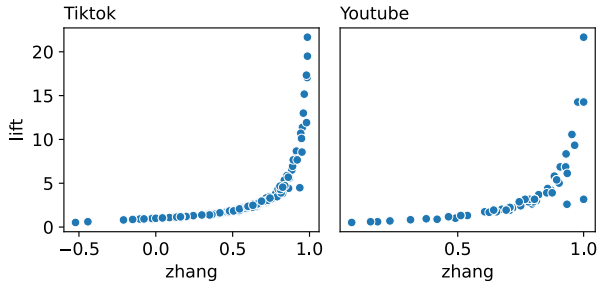


(a) Apriori

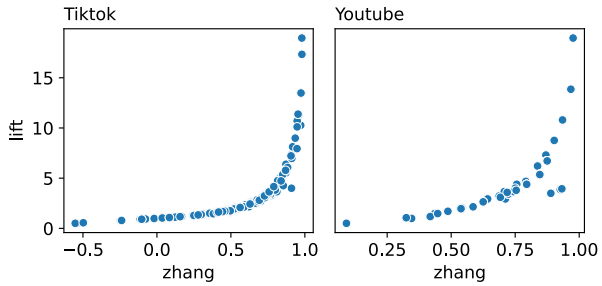


(b) HU-FIM

Fig. 5: Relationship between support and confidence to empirically set thresholds. Rules generally have lower support than confidence, the metric that measures the probability of one item following the other. Lift values of rules are indicated by the size and opacity of the circles. The lift metric measures how often two items appear together compared to how often two statistically independent items should appear together. At a certain level of confidence lift and support values of rules start increasing. We set low thresholds to retain enough items to compare in the following step. The two pattern mining algorithms produced similar results, lift values are generally lower with HU-FIM.



(a) Apriori



(b) HU-FIM

Fig. 6: Lift and Zhang values to set further thresholds. Although Zhang is a variation on lift, an expression of association, it simultaneously expresses disassociation. Zhang therefore also allows negative values. We want to exclude the rules in the lower part of the plots before the exponential part of the curve begins. In a way, Zhang allows us to 'stretch out' the range of low lift values and find the point where items start to disassociate. There does not seem to be a big difference in terms of pattern mining algorithms.

both apps. We limited this section to two observations that stood out to us while highlighting differences between the results of different mining algorithms.

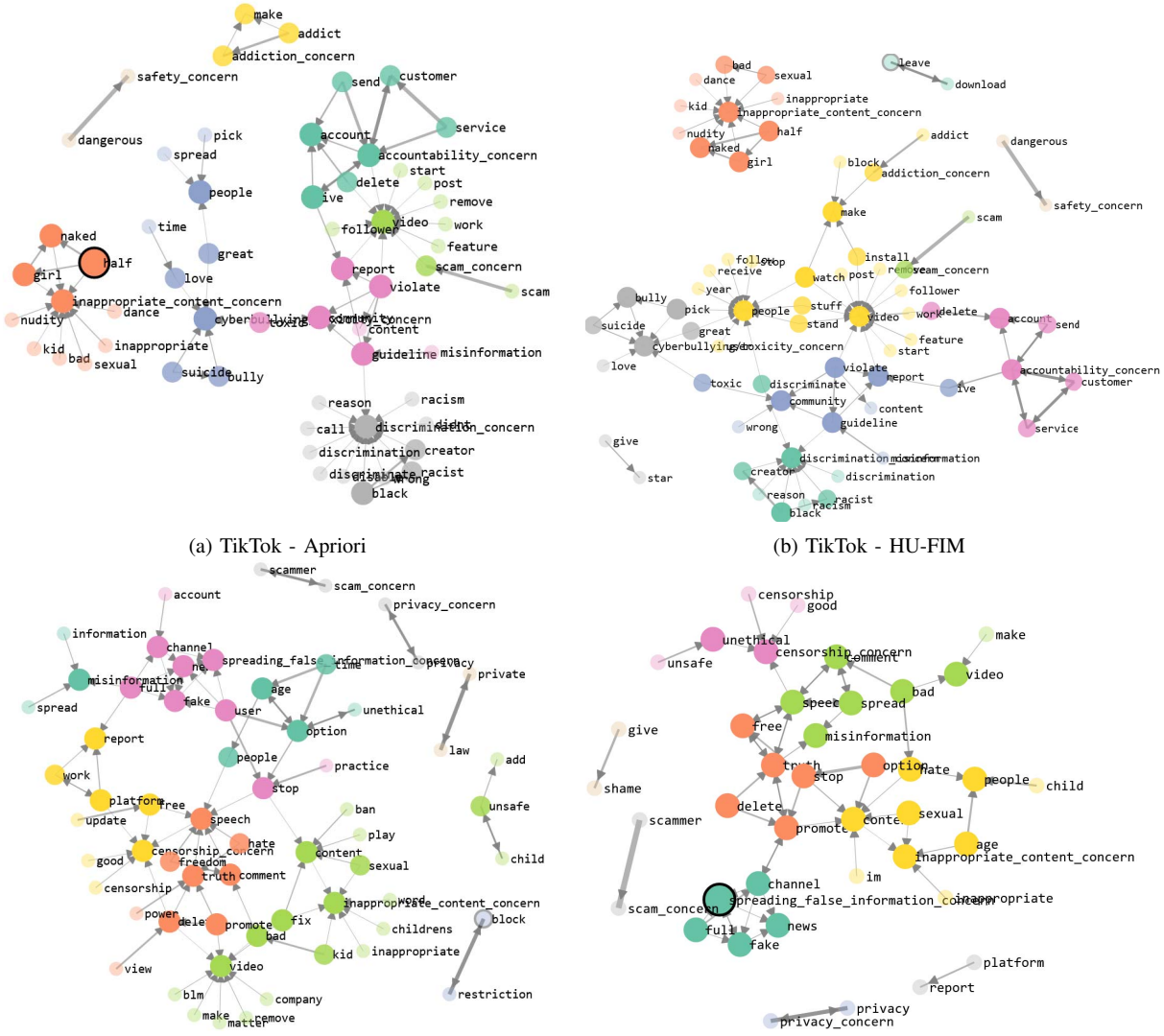
Inappropriate content was a recurring theme for both apps. For instance, in TikTok reviews (cf. Figures 7.a-b), we found an entirely disconnected subgraph of strong rules around the *inappropriate content* node. Here, the rules $\{half \rightarrow naked\}$ and $\{girl \rightarrow naked\}$ stand out among other patterns relating to the words *sexual*, *kid*, *dance*, and *nudity*.

We found the same subgraph in the high utility patterns, where, in addition, the rule $\{sexual \rightarrow bad\}$ transpired. We found similar patterns for YouTube in Figures 7.c-d, namely $\{sexual \rightarrow content\}$ for both Apriori and HU-FIM, in addition to the words *children* and *kid* for Apriori and *age* for HU-FIM that point at the concern.

We also found ethical concern patterns that are unique to individual apps. For YouTube, we found patterns surrounding *censorship* such as $\{free \rightarrow speech\}$ and $\{hate \rightarrow speech\}$. Additionally, using HU-FIM, we find $\{spread \rightarrow misinformation\}$ in the vicinity of *censorship*, instead of the *spreading false information* concern; the opposite is the case for Apriori.

App	Algorithm	Itemsets	Rules	Confident	Strong
TikTok	Apriori	155	310	159	73
	HU-FIM	165	330	166	77
YouTube	Apriori	143	286	156	89
	HU-FIM	62	124	68	37

TABLE III: Results per app and mining algorithm. The *Itemsets* column contains the total number of frequent itemsets, the following column contains the total number of subsequently generated association rules. The *Confident* column shows the number of rules remaining after setting the confidence threshold. The *Strong* column holds the final number of rules after applying lift and Zhang thresholds.



(a) TikTok - Apriori

(b) TikTok - HU-FIM

(c) YouTube - Apriori

(d) YouTube - HU-FIM

Fig. 7: Directed graph visualizations of the strong rules per app and algorithm. The nodes represent items; their size and opacity indicate the degree of incoming edges. Their color was determined by a clustering algorithm. Ethical concern label item node names have *_concern* appended. The edges point from antecedent to consequent; their weight is given by the rules' confidence. We read the graph by identifying chains of items from and to concern labels. We first observe the direct neighborhood, i.e., context, of a concern node. We then map out the shared context of two concern nodes by following chains between.

V. Discussion

Our preliminary results suggest that pattern mining and interactive graph visualizations are promising techniques for uncovering context, patterns, and overarching themes—within and among—ethical concerns about software expressed in app reviews. They could also be useful for highlighting patterns worthy of close inspection, detecting conflicting requirements, and uncovering sub-types of ethical concerns.

For instance, we uncovered that the context of the *inappropriate content* concern revolves around age-inappropriateness and sexual themes in both apps. Interestingly, these themes are captured in a disconnected subgraph for TikTok, showing a lack of shared context with other concerns. Both of these themes (age-inappropriateness and sexual content) are potential sub-types for the *inappropriate content* concern. Detecting users' shared concerns may be straightforward, yet we also uncovered patterns of potentially contradicting concerns surrounding *censorship* in YouTube reviews. For example, concerns related to *free speech* and *hate speech* can often be at odds since some users demand absolute free speech while others require the removal of hate speech. This is a complex

	Itemset		Association Rule		
App	Support	Utility	Confidence	Lift	Zhang
TikTok	0.02	4	0.4	2	0.5
YouTube	0.04	3	0.5	2	0.7

TABLE IV: List of set thresholds per app in the different stages of the approach. We set a support threshold on single items before generating 2-item sets. In addition, for HU-FIM we set a utility threshold per app. We further list the thresholds on confidence, lift, and Zhang values applied to the association rules that were generated from itemsets.

issue that we hope to shed further light on in the future. Understanding context is essential for prioritizing ethical concerns. For example, Figures 7.a-b show that *cyberbullying* is linked to suicide, highlighting the importance of addressing specific ethical concerns with urgency.

Despite considerable corpus size differences and overall higher thresholds, we found more strong rules for YouTube than for TikTok with Apriori. However, when we apply HU-FIM, we find less than half as many strong rules, which we consider a desirable effect. In its graph, we also find fewer nodes with only one edge. The number of strong rules for the bigger TikTok corpus remains stable between algorithms. The disparity could indicate differences in the review quality or language for both apps; we will investigate the issue in future work.

VI. Limitations

Our approach has limitations regarding corpus sizes, i.e., representativeness and possibly sampling biases introduced by user demographics, classifier overfitting, or researcher biases. On that note, we express our alignment with the ground truth training labels, which we reused from a previous study.

Furthermore, concerns can change over time in this quickly evolving field, adding a temporal limit to our data. Considering the components of our approach, we recognize that further potential itemset miners must be compared. We must evaluate results in terms of time efficiency and output relevancy.

Utility values were chosen arbitrarily, and we have not exhausted the interestingness metrics list. We cannot confirm that we have found the optimal interesting measures suitable for our approach. To adequately and consistently compare the results of various setups, we require a systematic strategy to read the context from graphs, which is the content of our future plans.

With regard to our data, we cannot entirely exclude the existence of bot-coordinated fake app reviews. None of the many closely read reviews appeared inorganic, but bot detection should be considered before classification.

Moreover, the number of reviews per concern class was imbalanced, leading to an unequal likelihood for patterns to arise around certain ethical concerns, such as addiction. Apart from collecting larger datasets, we may also be able to address this issue during itemset mining or interestingness thresholding.

We, the researchers of this study, set our own utility values and thresholds, reflecting our understanding of word utility and association rule interestingness. Consequently, what is deemed strong or interesting differs based on our perspectives, and may change when replicated by someone else. However, we believe that this study can serve as a valuable guide for future work as we provide a transparent report of our reasoning on threshold selection. Similarly, the reading of context from network graphs in this work was potentially influenced by our own biases.

VII. Future Plan

We plan to continue the evaluation of pattern recognition as a tool to uncover the thematic context of automatically classified user reviews. In the future, we will evaluate our approach on larger sets of data and more applications. We will also examine additional pattern mining algorithms.

So far, we manually thresholded association rules according to measures of interestingness. We will examine additional measures and write more comprehensive guidelines about the thresholding for software practitioners.

We used arbitrary intuition-based values of utility during HU-FIM, but we would like to determine a set of well-argued utility values in the future. We will also study patterns without removing stopwords, and rather decide on word inclusion in frequent itemsets by their utility value. Adding a dictionary of internet slang to value nouns and verbs correctly appears necessary too. Including the ethical concern type in the transaction led to promising results. We plan to add additional information such as sentiment, popularity, rating, and emotion to the transaction data. We also plan to compare results of temporal high utility-frequent patterns of ethical concerns in app reviews. This way, we can analyze the temporal relevancy of patterns about ethical concerns. To strategically interpret the patterns in the graph, we plan to find relevant chains of words connected to ethical concerns. To eliminate researcher bias during graph interpretation, we will devise a strategy for graph traversal and context extraction. Since node edges carry a value, chains of nodes can be valuated as a total. We will also validate the actual usefulness of our approach with software practitioners. Finally, we will continue to uncover end-users' ethical concerns about software and compile the required knowledge to build ethical software.

VIII. Conclusion

Our preliminary results show that the combination of pattern mining and network graph visualizations is effective for detecting patterns and giving context to ethical concerns about software. They could also be useful for detecting conflicting concerns, and uncovering sub-types of ethical concerns. Our approach could help software developers and researchers in enhancing software design by considering user's ethical concerns at a large-scale and in a more systematic manner. This work is a first step towards a large-scale, well-informed analysis of requirements about ethics in software, and, ultimately, more ethically conscious software.

REFERENCES

[1] A. Besmer, J. Watson, and M. S. Banks, "Investigating user perceptions of mobile app privacy: An analysis of user-submitted app reviews," *Int. J. Inf. Secur. Priv.*, vol. 14, pp. 74–91, 2020.

[2] P. Nema, P. Anthonysamy, N. Taft, and S. T. Peddinti, "Analyzing user perspectives on mobile app privacy at scale," in *Proceedings of the 44th International Conference on Software Engineering*, pp. 112–124, 2022.

[3] M. Tushev, F. Ebrahimi, and A. Mahmoud, "Digital discrimination in sharing economy a requirements engineering perspective," in *2020 IEEE 28th International Requirements Engineering Conference (RE)*, pp. 204–214, IEEE, 2020.

[4] L. Olson, N. Tjikhoeri, and E. Guzmán, "The best ends by the best means: Ethical concerns in app reviews," 2024.

[5] J. Li, Y. Zhang, and J. Mou, "Understanding information disclosures and privacy sensitivity on short-form video platforms: An empirical investigation," *Journal of Retailing and Consumer Services*, vol. 72, p. 103292, 2023.

[6] L. Olson, E. Guzmán, and F. Kunneman, "Along the margins: Marginalized communities' ethical concerns about social platforms," in *International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS)*, pp. 71–82, 2023.

[7] D. Pagano and B. Bruegge, "User Involvement in Software Evolution Practice : A Case Study," in *Proc. of the International Conference on Software Engineering*, pp. 953–962, 2013.

[8] L. Hoon, R. Vasa, J.-G. Schneider, J. Grundy, and Others, "An analysis of the mobile app review landscape: trends and implications," *Swinburne University of Technology, Tech. Rep*, 2013.

[9] D. Pagano and W. Maalej, "User feedback in the appstore: an empirical study," in *Proc. of the International Requirements Engineering Conference*, pp. 125–134, 2013.

[10] W. Martin, F. Sarro, Y. Jia, Y. Zhang, and M. Harman, "A survey of app store analysis for software engineering," *IEEE transactions on software engineering*, vol. 43, no. 9, pp. 817–847, 2016.

[11] E. Guzman, M. El-Halaby, and B. Bruegge, "Ensemble Methods for App Review Classification: An Approach for Software Evolution," in *Proc. of the Automated Software Enginering Conference*, pp. 771–776, 2015.

[12] W. Maalej and H. Nabil, "Bug report, feature request, or simply praise? On automatically classifying app reviews," in *Proc. of the International Requirements Engineering Conference*, pp. 116–125, 2015.

[13] S. Panichella, A. Di Sorbo, E. Guzman, C. A. Visaggio, G. Canfora, and 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)*, pp. 281–290, 2015.

[14] N. Chen, J. Lin, S. C. Hoi, X. Xiao, and B. Zhang, "Ar-miner: mining informative reviews for developers from mobile app marketplace," in *Proceedings of the 36th international conference on software engineering*, pp. 767–778, 2014.

[15] L. Villarroel, G. Bavota, B. Russo, R. Oliveto, and M. Di Penta, "Release planning of mobile apps based on user reviews," in *Proceedings of the 38th International Conference on Software Engineering*, pp. 14–24, 2016.

[16] A. Di Sorbo, S. Panichella, C. V. Alexandru, J. Shimagaki, C. A. Visaggio, G. Canfora, and H. C. Gall, "What would users change in my app? summarizing app reviews for recommending software changes," in *Proc. of the International Symposium on Foundations of Software Engineering*, pp. 499–510, 2016.

[17] L. V. Galvis Carreño and K. Winbladh, "Analysis of user comments: an approach for software requirements evolution," in *Proc. of the International Conference on Software Engineering*, pp. 582–591, 2013.

[18] C. Iacob and R. Harrison, "Retrieving and analyzing mobile apps feature requests from online reviews," in *Proc. of the Working Conference on Mining Software Repositories*, pp. 41–44, 2013.

[19] F. M. Kifetew, A. Perini, A. Susi, A. Siena, D. Muñante, and I. Morales-Ramirez, "Automating user-feedback driven requirements prioritization," *Information and Software Technology*, vol. 138, p. 106635, 2021.

[20] E. Guzman and W. Maalej, "How do Users like this Feature? A fine grained Sentiment Analysis of App Reviews," in *Proc. of the International Conference on Requirements Engineering*, pp. 153–162, 2014.

[21] X. Gu and S. Kim, "What parts of your apps are loved by users?," in *Proc. of the International Conference on Automated Software Engineering (ASE)*, pp. 760–770, 2015.

[22] F. Palomba, P. Salza, A. Ciurumelea, S. Panichella, H. Gall, F. Ferrucci, and A. De Lucia, "Recommending and localizing change requests for mobile apps based on user reviews," in *Proc. of the International Conference on Software Engineering (ICSE)*, pp. 106–117, 2017.

[23] H. Khalid, E. Shihab, M. Nagappan, and A. E. Hassan, "What do mobile app users complain about?," *IEEE software*, vol. 32, no. 3, pp. 70–77, 2014.

[24] Z. S. Li, M. Sihag, N. N. Arony, J. B. Junior, T. Phan, N. Ernst, and D. Damian, "Narratives: the unforeseen influencer of privacy concerns," in *2022 IEEE 30th International Requirements Engineering Conference (RE)*, pp. 127–139, IEEE, 2022.

[25] M. Shahin, M. Zahedi, H. Khalajzadeh, and A. R. Nasab, "A study of gender discussions in mobile apps," *International Conference on Mining Software Repositories*, 2023.

[26] D. Lewis, Y. Yang, T. Russell-Rose, and F. Li, "Rcv1: A new benchmark collection for text categorization research.," *Journal of Machine Learning Research*, vol. 5, pp. 361–397, 04 2004.

[27] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.

[28] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," *Proc. 20th Int. Conf. Very Large Data Bases VLDB*, vol. 1215, 08 2000.

[29] R. Uday Kiran, T. Yashwanth Reddy, P. Fournier-Viger, M. Toyoda, P. Krishna Reddy, and M. Kitsuregawa, "Efficiently finding high utility-frequent itemsets using cutoff and suffix utility," in *Advances in Knowledge Discovery and Data Mining* (Q. Yang, Z.-H. Zhou, Z. Gong, M.-L. Zhang, and S.-J. Huang, eds.), (Cham), pp. 191–203, Springer International Publishing, 2019.

[30] P.-N. Tan, M. S. Steinbach, and V. Kumar, "Introduction to data mining," pp. 327–414, 2005.

[31] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," *SIGMOD Rec.*, vol. 22, p. 207–216, jun 1993.

[32] S. Raschka, "Mlxtend: Providing machine learning and data science utilities and extensions to python's scientific computing stack," *The Journal of Open Source Software*, vol. 3, Apr. 2018.

[33] U. Kiran, "Pattern mining (pami): A python library containing several algorithms to discover user interest-based patterns in a wide-spectrum of datasets across multiple computing platforms.," https://github.com/UdayLab/PAMI (accessed on 2024-02-06).

[34] P. University. https://wordnet.princeton.edu/, 2010.

[35] G. A. Miller, "Wordnet: A lexical database for english," *Commun. ACM*, vol. 38, pp. 39–41, 1992.

[36] G. Piatetsky-Shapiro, "Discovery, analysis, and presentation of strong rules," in *Knowledge Discovery in Databases*, 1991.

[37] X. Yan, C. Zhang, and S. Zhang, "Confidence metrics for association rule mining.," *Applied Artificial Intelligence*, vol. 23, pp. 713–737, 10 2009.

[38] V. O. Taskesen E., ""d3blocks: A python package to create interactive d3js visualizations."," 2022. https://d3blocks.github.io/d3blocks (accessed on 2024-02-06).

[39] R. J. Bayardo and R. Agrawal, "Mining the most interesting rules," in *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, p. 145–154, 1999.