On the Violation of Honesty in Mobile Apps: Automated Detection and Categories

Humphrey O. Obie
HumaniSE Lab
Monash University
Melbourne, Australia
humphrey.obie@monash.edu

Idowu Ilekura
Data Science Nigeria
Lagos, Nigeria
ilekuraidowu@gmail.com

Hung Du
Applied Artificial Intelligence Inst.
Deakin University
Melbourne, Australia
hung.du@deakin.edu.au

Mojtaba Shahin
School of Computing Technologies
RMIT University
Melbourne, Australia
mojtaba.shahin@rmit.edu.au

John Grundy
HumaniSE Lab
Monash University
Melbourne, Australia
john.grundy@monash.edu

Li Li
Faculty of IT
Monash University
Melbourne, Australia
li.li@monash.edu

Jon Whittle
CSIRO's Data61
Melbourne, Australia
Jon.Whittle@data61.csiro.au

Burak Turhan
University of Oulu
Oulu, Finland
burak.turhan@oulu.fi

ABSTRACT

Human values such as *integrity, privacy, curiosity, security,* and *honesty* are guiding principles for what people consider important in life. Such human values may be violated by mobile software applications (apps), and the negative effects of such human value violations can be seen in various ways in society. In this work, we focus on the human value of *honesty*. We present a model to support the automatic identification of violations of the value of honesty from app reviews from an end-user perspective. Beyond the automatic detection of honesty violations by apps, we also aim to better understand different categories of honesty violations expressed by users in their app reviews. The result of our manual analysis of our honesty violations dataset shows that honesty violations can be characterised into ten categories: unfair cancellation and refund policies; false advertisements; delusive subscriptions; cheating systems; inaccurate information; unfair fees; no service; deletion of reviews; impersonation; and fraudulent-looking apps. Based on these results, we argue for a conscious effort in developing more honest software artefacts including mobile apps, and the promotion of honesty as a key value in software development practices. Furthermore, we discuss the role of app distribution platforms as enforcers of ethical systems supporting human values, and highlight some proposed next steps for human values in software engineering (SE) research.

MSR '22, May 23–24, 2022, Pittsburgh, PA, USA

ACM ISBN 978-1-4503-9303-4/22/05…$15.00
https://doi.org/10.1145/3524842.3527937

ACM Reference Format:
Humphrey O. Obie, Idowu Ilekura, Hung Du, Mojtaba Shahin, John Grundy, Li Li, Jon Whittle, and Burak Turhan. 2022. On the Violation of Honesty in Mobile Apps: Automated Detection and Categories. In *19th International Conference on Mining Software Repositories (MSR '22), May 23–24, 2022, Pittsburgh, PA, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3524842.3527937

1 INTRODUCTION

Human values such as *integrity, privacy, curiosity, security,* and *honesty*, are the guiding principles for what people consider important in life [11]. These values influence the choices, decisions, relationships, and the concept of ethics for people and society at large whether or not they are formally articulated in this terminology [60]. The relationship between human values and technologies is important, especially for ubiquitous technologies like mobile software applications (apps) [48]. Mobile apps are a convenience to modern society and have seen usage in carrying out both simple and complex tasks, from entertainment (e.g., video sharing apps) and health (e.g., fitness trackers) to finance (e.g., banking apps). End-users of these apps hold certain expectations influenced by their human values considerations, e.g., the privacy of data, transparency of processes in apps, and ethical behaviour of platforms and software companies [48]. The violation of these value considerations is detrimental to the end-user, software platforms, companies, and society in general [69].

Recent work on human values in software engineering (SE) based on the Schwartz theory of basic human values [60, 61] have mapped human values to specific ethical principles. For example, Perera et al. mapped values to the GDPR principles [54] and Winter et al. mapped values to the ACM Code of Ethics [71]. Other studies such as [48] have explored the violation of human values in mobile apps using app reviews as a proxy. The recent study by Obie et al. showed that the value of *honesty* (a sub-item of *benevolence* based on Schwartz theory [60]) is violated by mobile apps [48].

Honesty, often perceived to be a very important human value [41], describes a character quality of being sincere, truthful, fair, and straightforward, and refraining from lying, cheating, deceit, and fraud [15]. The importance of the value of honesty is clearly articulated in the ACM Code of Ethics: "Honesty is an essential component of trust. A computing professional should be transparent and provide full disclosure of all pertinent system limitations and potential problems. Making deliberately false or misleading claims, fabricating or falsifying data, and other dishonest conduct are a violation of the Code..." [21]. Nonetheless, there have been many flagrant violations of the value of honesty by mobile app platforms and software companies [17, 23, 67].

Consider the following example of the violation of honesty. The dating platform (Match.com) has been accused of faking love interests using bots and fake profiles to fool consumers into buying subscriptions and exposing them to the risk of fraud and other deceptive practices [56]. During a period of over three years, the company allegedly delivered marketing emails (i.e., the *"You have caught his eye"* notification) to potential consumers after the company's internal system already flagged the message sender as a suspected bot or scammer. The company also violated the "Restore Online Shopper's Confidence Act" (ROSCA) by making the unsubscription process tedious; internal documents showed that users need to make more than six clicks to cancel their subscription, resulting in the U.S. Federal Trade Commission (FTC) suing Match.com for "deceptive advertising, billing, and cancellation practices" [56].

Consider the more recent example of Shaw Academy who offered users a free trial to its online education platform and charged them a subscription fee even after they had cancelled before the end of the trial period and refused to refund the users [72]. The outcome of an investigation by the Australian Competition & Consumer Commission (ACCC) ordered the company to refund approximately $\$50,000$ to the affected users and pledge to improve their system [72]. Here is an example review of dubious charges to a user account for a calendar reminder app:

"I've been charged $45+ on 2 separate occasions in the month I've had the 'premium' version. It advertises $3.50 for a premium subscription but saw nowhere where it said they would make additional charges. There is absolutely no reason a calender reminder app should charge this much without telling you or without being deceptive."

Other examples include companies deliberately hiding data breaches from the authorities and customers [7, 63]. These violations of the value of honesty result in decreased trust from users, poor uptake of apps, and reputational and financial damage to the organisations involved. This also emphasises the need to consider human values more proactively in software engineering practice.

To detect the violation of the value of honesty in mobile apps, we utilised user's comments expressed in app reviews, as reviews are a valuable resource and have been shown to be a proxy for detecting users' challenges and requirements [5, 14, 22, 48, 64]. To this end, we formulated the identification of the violation of honesty in app reviews as a classification problem. We trained and compared five machine learning models based on a manually annotated dataset to learn the features that are representative of the violation of honesty in app reviews. The best performing model (Support Vector Machine) has an F1 score of 0.89, a precision of 0.94, and a recall of 0.84. Additionally, beyond the automatic detection of honesty violations, this work also aims to understand the different categories of honesty violations expressed in app reviews. Thus, we manually analysed reviews containing honesty violations. Our resulting taxonomy shows that honesty violations can be characterised into ten categories: **unfair cancellation and refund policies, false advertisements, delusive subscriptions, cheating systems, inaccurate information, unfair fees, no service, deletion of reviews, impersonation,** and **fraudulent-looking apps**. In summary, this work makes the following contributions:

- We present machine learning models and datasets to aid the automatic detection of the violation of the human value of honesty in reviews. Our publicly available replication package supports researchers and practitioners to adapt, replicate, and validate our study [6].
- We provide insight into the different categories of honesty violations prevalent in app reviews by creating a taxonomy based on a manual analysis of the honesty violations dataset.
- We present a set of practical recommendations and future research directions to deal with the challenges of the violations of the human value of honesty in apps that would benefit end-users and society.

2 RELATED WORK

2.1 Mining App Reviews

Several works have been carried out to provide insights into user reviews and how these reviews can aid software professionals in app maintenance [9, 53, 62] and evolution [12, 35, 38, 51]. Guzman and Maalej adopted NLP techniques to locate fine-grained app features in reviews with the aim of supporting software requirements tasks [22]. A related work utilised Latent Dirichlet Allocation (LDA) technique and linguistic rules to group feature requests from users as expressed in their reviews, and the results from this study showed that users care about frequent updates, improved support, more customisation options, and new levels (for game apps) [25].

Some studies have focused on the automatic classification of app reviews into useful categories. To aid software professionals in prioritising accessibility issues, AlOmar et al. developed a machine learning model for identifying accessibility-related complaints in app reviews [5]. Panichella et al. introduced a taxonomy for classifying app reviews and using a combination of NLP and sentiment analysis classified app reviews into their proposed taxonomy [52].

Other works have introduced tools to support the extraction of insights from app reviews. For example, Vu et al. proposed MARK, a keyword-based tool for detecting trends and changes that relate to occurrences of serious issues in reviews [58]. Similarly, Di Sorbo et al. introduced SURF, a tool that condenses thousands of reviews into coherent summaries to support change requests and planning of software releases [14].

The above studies show that app reviews are a useful resource for gathering requirements, detecting issues, and more generally for supporting software professionals in evolving their apps. This work also aims to support app maintenance and evolution by effectively detecting potential violations of the value of honesty from the user's perspective in app reviews. In addition, it would aid software professionals in delivering software products that build trust in

users, as the honesty (real or perceived) of companies can affect how users engage with with their products [73].

2.2 Human Values in Software Engineering (SE)

Human values are enduring beliefs that a specific mode of conduct or end state of existence is personally or socially preferable to an opposite or converse mode of conduct or end state of existence [59]. Human values have been well-studied in the social sciences and have begun to see adoption in other fields including design [4] and software engineering (SE) [34, 43].

The study of human values in SE is a relatively nascent line of research [44, 55] and is mostly based on the widely accepted and adopted Schwartz theory of basic human values [60, 61]. The Schwartz theory is built on a survey conducted in over 80 countries covering different demographics. This theory categorises values into 10 broad categories, namely: *self-direction, stimulation, hedonism, achievement, power, security, conformity, tradition, benevolence,* and *universalism*. These 10 categories in turn are made up of 58 value items, e.g., the value category of ***benevolence*** covers the value items of ***honesty***, *responsible, helpful, forgiving, loyal, mature love, a spiritual life, meaning in life,* and *true friendship* (c.f [60]). However, our focus in this work is on the value item of *honesty*, based upon the prevalence of the value category of *benevolence* in prior research [48], the recent cases of the violations of *honesty* by companies in the media, e.g., [56, 72], and the need to understand this phenomenon more closely in SE.

Studies in the social sciences have investigated the value of (dis)honesty at the individual and organisational levels [19], and the policy implication of dishonesty in everyday life [40]; while others have explored the motivation for dishonest behaviours [1] including students in classroom settings [30] and workers in crowd-working environments [26]. Keyes argues that euphemising the violation of the value of honesty desensitises people to its implications and consequences in society [28].

However, within the context of SE, Whittle et al. argued that software companies need to consider human values in the development of software systems and make them "first-class" entities throughout the software development life cycle [69]. Another study made a case for the evolution of current software practices and frameworks to embed human values in technology instead of a revolution of the SE field [24].

Another line of research considered methods for measuring human values in SE. For example, Winter et al. introduced the Values Q-sort instrument for measuring human values in SE [71]. Applying the Values Q-sort instrument to 12 software engineers resulted in 3 software engineer values "prototype". Similarly, Shams et al. utilised the portrait values questionnaire (PVQ) to elicit the values of 193 Bangladeshi female farmers in a mobile app development project [65]. The result of the study showed that conformity and security were the most important values while power, hedonism, and stimulation were the least important. More recently, Obie et al. argued that the instruments for eliciting and measuring values should be contextualised to specific domains [49].

Recent studies have adopted the use of app reviews as an auxiliary data source for eliciting values requirements. Shams et al. analysed 1,522 reviews from 29 agricultural mobile apps to understand the values that are both represented and missing from these apps [64]. Obie et al. proposed a keyword dictionary-based NLP classifier to detect the value categories violated in app reviews [48]. The results of the application of the classifier to 22,119 reviews showed that benevolence and self-direction were the most violated categories while conformity and tradition were the least violated.

The studies highlighted have been instrumental in pushing the frontiers of human values in SE, and the closely related works such as [48, 64] have provided insights to violations of value categories. Our work complements these by zooming in on a specific value item; ***honesty*** (within the most violated category of ***benevolence*** [48]), to provide a more nuanced understanding of its violations. In addition, we provide a taxonomy of the different categories of honesty violations in reviews to better understand how the violation of the value of honesty is reported. We hope that other researchers would be encouraged to investigate other specific value categories, and more generally explore the field of human values in SE.

Table 1: Statistics of the dataset.

Number of Apps	713
App Categories	25
All Reviews	236,660
Honesty-related Reviews (after keywords filter)	4,885
Honesty Violation Reviews (after manual validation)	401

3 RESEARCH DESIGN

Our goal in this study is to automatically identify reviews discussing honesty values and indicating from these reviews to determine the different types of honesty violations documented. To do this we define the following research questions (RQs):

RQ1. *Can we effectively identify reviews documenting honesty violations automatically?*

RQ2. *What types of honesty violations are reported in these app reviews?*

3.1 A Dataset of Honesty-related Reviews

The first step to answering our RQs is creating a dataset of user reviews documenting perceived honesty violations by apps.

3.1.1 Data Collection. To build this dataset, we collected a total of 236,660 reviews - 214,053 reviews from the public dataset of Eler et al. [18], and an additional 22,607 reviews from the public dataset of Obie et al. [48]. These reviews were collected from a total of 713 apps in 25 categories. The apps and reviews were intended to cover a diverse range of categories and audiences. Table 1 summarises the statistics of our combined app review dataset. Our dataset can be found here [6].

3.1.2 Data Labeling. Given the sheer size of the dataset and the manual labour required to label the dataset, we used two approaches to label the 236,660 reviews: a keyword-based approach and manual labeling. We first adopt a set of keywords to filter the 236,660 reviews to include those related only to the value of honesty. These keywords are based on the dictionary of human values created by Obie et al. [48]. The set of keywords comprise a total of 48 words

semantically related to honesty. The keywords are available in [6]. After applying this keyword filter, the number of reviews was reduced from 236,660 reviews to 4,885 potential candidate honesty-related reviews (we call these 4,885 reviews ***honesty_potential*** reviews).

However, adopting a keyword-based approach is error-prone and may result in a lot of false positives. Hence, we manually analysed the ***honesty_potential*** reviews to exclude the false positives. Moreover, the application of keywords filter and subsequent manual analysis check have been applied in recent studies [5, 18].

The ***honesty_potential*** reviews were labelled and validated in 25% increments in the following manner. The first analyst labelled the first 25% percent of the ***honesty_potential*** reviews to determine which of the reviews contain the violation of the value of honesty as perceived by the user in the review. The second analyst validated the outcome. The disagreements were resolved in a meeting using the negotiated agreement approach to address issues of reliability [8, 42]. Then the next 25% were labelled by the first analyst, validated by the second analyst, and disagreements resolved in a meeting as in the first round. The same procedure was repeated for the third and fourth rounds of the labelling process. Also, the labelling and validation were done over eight weeks to avoid fatigue. Based on our manual labelling, we found that out of the 4, 885 filtered reviews (the ***honesty_potential*** reviews), only 401 were honesty violations reviews, i.e., true positives. We refer to these 401 honesty violations reviews as ***honesty_violations*** reviews.

Next, we randomly selected 401 reviews from the remaining 4,484 ***honesty_potential*** reviews (4,885 ***honesty_potential*** reviews - 401 ***honesty_violations*** reviews). We refer to these 401 reviews, which contain honesty-related keywords (but not violations), as ***honesty_non_violations*** reviews. We used a total of 802 reviews: 401 ***honesty_violations*** and 401 ***honesty_non_violations*** reviews to build a balanced dataset called **honesty_discussion** dataset for training and evaluating machine learning models in Section 4. We note here that using the manually validated false-positive *honesty_non_violations* reviews is important for machine learning models. It is because these reviews include certain keywords syntactically related to honesty but semantically irrelevant to honesty violations - an important difference we want our models to learn. In summary, the **honesty_discussion** dataset consists of 802 reviews: 401 ***honesty_violations*** reviews and 401 ***honesty_non_violations*** reviews. Other studies have used similar numbers of text documents in classification tasks [32, 33].

4 AUTOMATIC CLASSIFICATION OF HONESTY VIOLATIONS (RQ1)

4.1 Approach

Manually classifying honesty violations in app reviews is challenging for practitioners because it is error-prone, labor-intensive, and demands substantial domain expertise. Hence, an automated approach is required to recognise honesty violations in app reviews. This research question aims to develop machine learning models to differentiate between honesty and non-honesty reviews automatically. The machine learning models are applied on the 802 **honesty_discussion** dataset which consists of 401 ***honesty_violations*** reviews and 401 ***honesty_non_violations*** reviews.

4.1.1 Data Preparation. We applied some common techniques to remove possible noise from the **honesty_discussion** dataset. This step was needed so a learning model can classify reviews correctly. To achieve this, we applied natural language processing techniques such as removing capitalisation, removing emojis, tokenising, removing stop words, and removing punctuation.

Case Normalisation: is the process of transforming original review texts into their lower case. This type of text cleansing helps us avoid repeated features of the same words with different font cases (e.g., "Honesty" and "honesty"). Furthermore, converting the text into its lower case does not affect its context as well as the users' expressions in our scenario.

Emoji Removal: Emojis are icons or a few Unicode characters that allow users to convey ideas, concepts, and emotions. If emojis are not carefully preprocessed, they can potentially affect the performance of a model in terms of accuracy. Hence, we removed Emoji from the review texts.

Tokenisation: is the process of splitting each original text into a set of words that do not contain white space. We divided apps reviews into their constituent set of words.

Stop-Word Removal: Stop words such as *is*, *am*, *are*, *for*, *the*, and others do not contain conceptual meaning of a review and create noise for a classification model. Removing stop words from the review texts helps us avoid repeated features of the same phases (e.g., "the bank account" and "bank account"). In our experiment, we used a comprehensive set of stop words that are well-known to the natural language processing community[1].

Punctuation Removal: We observed many reviews in the data collection containing punctuation such as *"..., ??, :(,"* and others that do not significantly contribute to a classification model. Hence, we removed punctuation from the app reviews.

4.1.2 Feature Extraction. After cleansing and preprocessing the dataset, we converted the app reviews in the dataset into their vector representation by using the pre-trained Bidirectional Encoder Representations from Transformers model [13], so-called BERT[2]. This is a language representation model trained on the BooksCorpus with 800 million words [74] and English Wikipedia with 2.5 billion words. The model receives a sequence of words as input and outputs a sequence of vectors. The model converted the review texts with different words into 768-dimensional vectors used as an input in a machine learning model. Each of these vectors is estimated by the average of embedded vectors of its constituent words. For instance, given a review text s that consists of n-words, $s = (w_1, \ldots, w_n)$, then, $\vec{s} \approx \frac{1}{n}(\vec{w_1} + \ldots + \vec{w_n})$, where $(\vec{w_1} + \ldots + \vec{w_n})$ are the embedded vectors of $(w_1, \ldots, w_n)$. Furthermore, these vectors capture both a semantic meaning and a contextualised meaning of their corresponding app reviews.

4.1.3 Model Selection and Tuning. Selecting a classification model that yields the optimal result is challenging. We selected five models, such as Support Vector Machine (SVM), Decision Trees (DT), Neural Network (NN), Logistic Regression (LR) and Gradient Boosting Tress (GBT) that are commonly used for text classification in

[1]The stop words can be accessed at https://gist.github.com/sebleier/554280#gistcomment-3126707

[2]The pre-trained BERT uncased model can be downloaded at https://huggingface.co/bert-base-uncased.

the natural language processing community [2]. Below is a brief description of each classification model used in our work.

Logistic Regression (LR) is a linear classifier. The data is fitted into a logistic function that generates the binary output such as 0 (i.e., an honesty_non_violation app review) or 1 (i.e., an honesty violation app review) based on probability.

Support Vector Machine (SVM) [46] is a classifier that finds hyperplane(s) in N-dimensional space (i.e., the number of features), which can further distinguish the data into multiple categories.

Decision Trees (DT) is one of the ensemble learners that builds trees for classification. Each tree represents a particular characteristic of the data. Given a 768-dimensional vector representation of a particular review text, DT classifies the review text into the category selected by most trees.

Gradient Boosting Trees (GBT) is one of the ensemble learners that builds trees and boosts them for classification. When a new tree is created, it corrects errors of previous trees fitted on the same provided data. This repeatedly correcting errors process is known as the boosting process. In addition, the gradient descent algorithm is used for optimisation during the boosting process. Thus, the method is called gradient boosting trees. The model classifies app reviews into a category based on the entire ensemble of trees.

Neural Network (NN) is a multilayer perceptron model which contains a set of interconnected layers where each layer contains a finite number of nodes. Each neural network architecture has one input layer, at least one hidden layer, and one output layer. The input data is transformed layer by layer via the activation function(s). During the training process, optimisation techniques such as stochastic gradient descent are used to optimise the performance of the model. The classified category of a particular app review is the collected result from the output layer.

Finding the hyperparameters for models to generate the optimal results is known as the fine-tuning process. We use grid search cross-validation to perform an exhaustive search to find the best set of hyperparameters for each classifier. To reproduce our results, we provide the selected hyperparameters for each selected model and the open-source GitHub repository in [6].

4.1.4 Cross Validation. To estimate the variance of the performance for each classification model, we used a 10-fold cross-validation technique. Here, we split the dataset in Section 3.1 into 10 chunks of data that contains an equal number of app reviews. Then, we perform the evaluation process where the training dataset contains 9 chunks of data, and another chunk of data is used as the testing dataset. Note that this is repeated until each chunk of data has been used as the testing dataset once. This approach helps us understand how well our selected models perform on unseen data.

4.2 Results

In this section, we report the results of our experiment evaluating the performance of the different machine learning models. We adopted the generally accepted metrics of ***accuracy, precision, recall***, and ***F1 score*** for this purpose. Other metrics such as the Matthews Correlation Coefficient (MCC) and confusion table are shown in Table 2.

We note here that all of the models performed well (with F1 scores of 0.79 and above).

Table 2: Comparison of confusion matrix and Matthews correlation coefficient (MCC) of classification models.

	SVM	LR	NN	RF	GBT
True negative	0.432	0.407	0.358	0.371	0.358
True positive	0.457	0.469	0.482	0.420	0.420
False positive	0.025	0.049	0.099	0.085	0.099
False negative	0.086	0.074	0.062	0.124	0.124
MCC	0.785	0.753	0.676	0.581	0.555

Table 3: Comparison of classification models.

	SVM	LR	NN	RF	GBT
Accuracy	0.889	0.877	0.840	0.790	0.778
Precision	0.949	0.905	0.830	0.829	0.810
Recall	0.841	0.864	0.886	0.773	0.773
F1 score	0.892	0.884	0.857	0.800	0.791

Table 3 shows the results of 5 different machine learning classification algorithms. The SVM algorithm came out to be the best performing model with an accuracy of 0.88, precision of 0.94, recall of 0.84, and an F1 score of 0.89. The second-best performing algorithm is the LR model, with an accuracy of 0.87, precision of 0.9, recall of 0.86, and an F1 score of 0.88.

Furthermore, the high performance of our SVM model makes it useful in practical applications for detecting the violation of the value of honesty in reviews.

4.2.1 Comparison with Baselines. One of the aims of our work is to introduce an automatic method for detecting honesty violations reviews that performs better than current approaches. Similar studies on text classification have compared their approaches to either the current state-of-the-art or a baseline random classifier [5, 39]. Hence we compare our best-performing machine learning model (SVM) with a baseline random classifier only since there is no current state-of-the-art in detecting the violation of honesty in app reviews, similar to what recent works have done [5, 39].

We used the statistics of our dataset to compute the metrics of the random classifier. The precision of a random classifier can be computed by dividing the number of honesty violation reviews by the total number of reviews:

$$precision = \frac{401}{236,660} = 0.0017$$

The recall is 0.5, as there are only two outcomes for a review classification: honesty violations reviews or honesty_non_violations reviews, with a 0.5 probability of a review containing the violation of the value of honesty. Based on the precision and recall values, we compute the F1 score of the baseline random classifier as:

$$F1\ score = 2 * \frac{0.0017 * 0.5}{0.0017 + 0.5} = 0.0034$$

Table 4 summarises the comparison of our best-performing machine learning model (SVM) with the baseline. As can be seen, the SVM model has a better performance than the baseline random classifier. Our SVM model has an F1 score of 0.89, while the baseline random classifier has F1 score of 0.0034, respectively. Table 4 also

Table 4: Comparison of our model to a baseline classifier.

	Our (SVM) approach			Random classifier		
	Precision	Recall	F1	Precision	Recall	F1
Classification	0.949	0.841	0.892	0.0017	0.5	0.0034
Improvement	-	-	-	558.235x	1.682x	262.353x

shows that our SVM model surpasses the baseline random classifier by 262.353 times in detecting honesty violations reviews.

RQ1 Answer: The SVM model surpasses the baseline random classifier in identifying the violation of the value of honesty in reviews. Our model achieves an F1 score of 0.892 with an improvement of 262.353 times the baseline random classifier in classifying honesty violation reviews from honesty_non_violation reviews.

5 CATEGORIES OF HONESTY VIOLATIONS (RQ2)

5.1 Approach

While the machine learning models in Section 4 could effectively distinguish between honesty violations reviews and honesty non-violations reviews, we are also interested in understanding the types of honesty violations reported in reviews. To this end, we applied the open coding procedure [20] on the 401 ***honesty_violations*** reviews. As discussed in Section 3.1, these reviews include honesty violations. First, an analyst followed the open coding technique to label all these 401 reviews and identified 10 types of honesty violations. The 401 ***honesty_violations*** reviews were assigned to these 10 categories. The results of the open coding were stored in an Excel spreadsheet file and shared with the second and third analysts. Then, the second analyst cross-checked the first 100 labelled reviews while the third analyst cross-checked the remaining 301 labelled reviews. Next, the first analyst held Zoom meetings with the second and third analysts to discuss and resolve the conflicts and disagreements. Note that the disagreements were resolved using the negotiated agreement approach [8, 42].

5.2 Results

Our analysis of the 401 ***honesty_violations*** reviews revealed 10 categories of honesty violations reported in app reviews. Below we provide a definition of these categories, sample reviews, and a summary of their prevalence. While we highlight the different categories within the violation of the value of honesty and provide example reviews, we note that the categories are not mutually exclusive. Table 5 shows these categories and the frequency of the corresponding reviews per category.

5.2.1 Unfair cancellation and refund policies. This category covers all reviews where the users perceive the cancellation and refund policy as unfair, nontransparent, or deliberately misleading. It also includes situations where the user feels that the developers deliberately make it difficult for the user to cancel their subscription. For example, in some apps, the user can sign up for a subscription with the click of a button within the app but cannot cancel the subscription from within the app; the user is asked to log in to a website to cancel the subscription. In other cases, the cancellation instruction is not clear and leads to a loop of cancellation steps. Examples of reviews claiming these practices include:

> 👎 *"The app allows you to accidentally sign up to premium with a push of a button. When you want to cancel, however, you can't do that via the app... You have to go to the webpage, enter details and cancel there."*

> 👎 *"Deceptive billing practices - information on cancelling is circular; emailed a link that advises to email. [It] doesn't have colour tag functionality across web and app; very poor UX and worse customer service."*

Sometimes, the app also makes it easy for the user to mistakenly activate a premium subscription in the way the interface and flow are designed, e.g.:

> 👎 *"Use with caution. It's unscrupulous about signing you up for a subscription when you're skipping past the in-app ads. It's not made clear once you've subscribed, and there's no way of cancelling it through the app."*

Another aspect of this category focuses on situations where the user perceives the refund steps and policies to be dishonest and unfair. This also involves situations where the refund policy does not cater to accidental subscriptions, e.g.:

> 👎 *"DO NOT SIGN UP FOR FREE TRIAL! IT IS A SCAM. YOU WILL GET CHARGED ANYWAY, AND YOU WILL NEVER GET YOUR MONEY BACK!! Once again, after numerous attempts to blame Google, this developer has still not refunded my $38. Once again, I cancelled 3 full days before the free trial ended but was still charged. Once again, [I] contacted the developer, who told me that I would receive a full refund within 7 to 10 days, and still nothing. I have saved the email, pricing this to be true. DO NOT TRUST THIS DEVELOPER. SCAM!!!!"*

5.2.2 False advertisements. This category relates to situations where the user perceives that the advertised features and functionalities of the app as described by the developers are not contained in the app. The user downloads the app or pays for a subscription on the basis of accessing certain functionalities or features only to find out the descriptions, including screenshots on the app distribution platform is different from the actual functionalities available in the app. Two examples of these are shown below:

> 👎 *"Couldn't find Google Assistant integration anywhere. Even though it's been advertised everywhere when searching the web for the app... It's even in the description of the app here. That's false advertising. I will edit my review when it's out of Beta and working in the final version."*

> 👎 *"The app doesn't listen to the watch at all. I've tried completing and snoozing and it does nothing. The watch app can only add tasks, so the screenshots they're sharing here are DECEPTIVE."*

In some cases, the app lures users into downloading the app on the basis that it is free-for-use only for the user to find out that the free-for-use is a trial version for a specific time period and not perpetually free as implied in the app description:

> 👎 *"The actual free version doesn't allow you anything, not even to learn how to use the app properly. That role is filled by 7 days of free premium. The free, on the description, is a lie. Is a paid-only app*

Table 5: Frequency (f) of app reviews in the honesty violation categories (out of 401 total *honesty_violations* reviews – note that some reviews fall into multiple categories).

	Unfair cancellation and refund policies	False advertisements	Delusive subscriptions	Cheating systems	Inaccurate information	Unfair fees	No service	Deletion of reviews	Impersonation	Fraudulent-looking apps
f	48 (12%)	55 (14%)	33 (8%)	93 (23%)	15 (4%)	106 (26%)	64 (16%)	6 (1.5%)	9 (2%)	29 (7%)

with temporary free access to its full features that gets practically useless after the 7-day trial... I don't like to be lied to."

In addition, the app developers (through the app description) make promises to users to give them certain benefits like a free premium subscription when a particular action is carried out (e.g., inviting a particular number of friends to sign up). However, they never truly fulfil their promises when the user fulfils their end of the bargain. These unfulfilled obligations are perceived by the end-user as a violation of honesty, e.g.:

👎 *"I love this app however I sent the link to several friends and they got the app and I received no premium time whatsoever. Don't be dishonest with your apps. That's lame."*

Another example relates to scenarios where the user is invited to make certain commitments based on a future reward and the developers bail out on their prior commitment:

👎 *"Shame on Them! Liars. I paid for the season pass TWICE (ONCE for my apple device and the other for my Samsung Device). I was falsely promised access to ALL FUTURE CONTENT. Now they are trying to charge me for the Parisian Inspired TOKENS! HOW DARE THEY LIE AND BAIT AND SWITCH."*

5.2.3 Delusive Subscriptions. Any review describing complaints related to unfair or nontransparent automatic subscription processes is classified under this category. There are instances where no notifications are provided to let the user know they are subscribed to the app or premium version of the app, and the user only finds out about the subscription from the deductions in their bank accounts:

👎 *"I just realised that I have been charged for some crappy premium service fee which I had no idea about when using the app. Why is this charge by default? Why was I not informed in the first place? Beware of scam for useless monthly premium fees!"*

👎 *"I can't believe I was charged 55.99. What are you giving me? Gold? I unsubscribed but saw mysterious charge in my bank account."*

Additionally, there is the issue of lack of user consent in the subscription process where certain apps do not provide a confirmation mechanism that prevents accidental subscriptions by the user, e.g.:

👎 *"Made me pay 1 year worth of subscription without my confirmation. Only used its free trial because I had to use it once. What a scam..."*

In some scenarios, the automatic subscription is hidden behind an in-app ad/feature, and an unsuspecting user who clicks on the feature is automatically subscribed to the premium version of the app without a clear warning or confirmation, e.g.:

👎 *"Deceptive practices. If you click the in-app "ad" that simply says enable notifications, you'll automatically be signed up and billed for their premium service. This bypasses the Google/Apple stores subscription model and bills your card directly. Not to mention it's impossible to downgrade from this service in the app itself; you have to visit their website, which is a deliberately obstructive hurdle considering you can upgrade in the app just fine."*

5.2.4 Cheating systems. All reviews concerning the user's perception of fraud by other persons or cheating within the inner workings of the app are classified under this category. Users complain of unfairness in either the process or outcome of the app, especially processes/outcomes that are supposedly statistically random. While accusations of this kind from the users are prevalent and subjective, they may not realistically be the case. However, we labelled these kinds of reviews based on the *perception* of the users as captured in their comments. Reviews related to this category are mostly found in games or game-like systems. For example:

👎 *"This game cheats. It uses words not found in the dictionary. Also it told me a word was unplayable, but it was the first best word option."*

👎 *"I play it with my sister often. However, there is the problem of the game and AI cheating. I rolled a 2 and a 3 at the start of the game and it moved me FOUR spaces forward not five. Four. That happened several times and I can assure you I was looking everytime it happened. I am very disappointed at the fact this game is cheating..."*

In some of the reviews, users complain that the game works properly when the user loses and parts with money and only freezes when the AI system in the app is about to lose. Based on the reviews, the users seem to be using real money in the games/apps. This complaint is a recurring theme within this category:

👎 *"You have to pay for it, then the game just freezes when you win against the CPU? Reset it over and again, keeps freezing unless it rolls something to not land on my property. Also, is the dice rigged against the CPU? Honesty? With as much as I owned in the beginning, none of the 3 CPUs would land on anything I owned. Anytime the last CPU needs to raise money, game freezes, guess ya just can't win."*

👎 *"there's a glitch in it that freezes the game from continuing when you're winning. The dice just disappears, but the trains and clouds and aircrafts keep moving. It's like It is designed so that one doesn't win them."*

👎 *"When playing against the computers when you're about to win and bankrupt the final computer the game conveniently freezes. It does not allow you to win. Not a very fun game to play, I want my money back."*

We consider this category important as some of these apps require the use of real money to play or for in-app purchases. If apps are dishonest in the underlying process of the systems that are expected to be fair, then that constitutes not only the violation of the value of honesty, it might potentially be a crime. This is worth considering, especially when the exact issue is raised by several users:

👎 *"Although you say that the dice is random, i cannot help but feel that it is rigged. Take a look at your reviews, there are many other players that feel the same. Can't be all of us are wrong. Or maybe we are suffering from mass hysteria?"*

Other non-games examples include cases where the user reports not having the full value of the fee they were charged for the app and feels cheated. For instance:

👎 *"Whenever I pay for parking the app always steals 5 minutes off my parking time. For example, I pay for 60 minutes and the timer starts at 54 minutes and 59 seconds. I am very upset, this has been happening for a while and probably to many more people as well. That is a lot of money!"*

👎 *"This app will not give you re requested amount of parking time. If you park for 15 minutes it will immediately say you have 11 minutes left. I understand that you have to charge but at least give me the requested amount of parking time."*

5.2.5 Inaccurate information. This category covers where users perceive that the app provides false or inaccurate information as captured in their reviews. This includes situations where inaccurate information can increase the likelihood of the user inadvertently making wrong selections at a cost to them. In the review below, the user complains the design of an app feature tricks them into paying for the wrong parking spot:

👎 *"When you need to pay for additional time, and click 'Recent' to pay for the most Recently parked in place - the first item is not the place you just parked in so it tricks you into paying for the wrong place (dark pattern). Please make the Recent accurately reflect the most recently parked in place."*

Another example review in this category is quite severe as it relates to a health emergency app providing potentially inaccurate information that might be detrimental to the user:

👎 *"Try to use this in an actual emergency and you'll just end up as a dead idiot holding a cellphone. The information is either useless or completely false in most cases. Don't bother downloading."*

Other less severe but important reviews where the user perceives the app provides inaccurate information or notification are shown below:

👎 *"Do not buy unless you are sure you want to. You will NOT be able to get it set up and working within the 15 minute refund window. The instructions online are so cryptic it (and wrong)."*

👎 *"Very annoying every time when you open the app it shows you have a notification. Then checking your notifications you don't have any."*

5.2.6 Unfair fees. This category relates to issues surrounding what the user considers to be unfair fees or charges. This also applies to cases where the user feels that they have not received a fair deal or that the app charges more money than it ought to. Because the definition of honesty also covers fairness, we also consider these kinds of issues a potential violation of the value of honesty. In the example below, the user complains of being charged more than they think is fair; they were charged a car parking rate for parking a bike.

👎 *"Went through the sign up process and parked my bike in a bike parking zone. Put in the correct zone details for the bike parking area and got charged a car parking rate. Rang support and they said there is no bike parking at that location. I explained there was and they told me to ring the council."*

Other examples of fees considered by the user to be unfair are:

👎 *"The app charges you 0.25 per transaction. So I paid 0.75 to pay for parking it charged me 0.25 service fee then I extended my parking 0.25 and it charged me again 0.25!!! Biggest scam in the world."*

👎 *"The only annoying things are that I have to buy any extra Monopoly Board in the same game when I already paid the main game. Can you not give extra Monopoly Boards in the same game for free. You are not fair!"*

This category is also reflected in the form of hidden charges where the user is not aware of subsequent charges made to their account. These hidden charges can take the form of a vague bill (as shown in the review below) or not notifying the user with respect to extra charges.

👎 *"This is a notorious company with horrible app I've ever used. They hide the history and details very deep for you to check and trace. And the monthly bill is also vague. I experienced they secretly bill me!"*

👎 *"LOOK OUT PEOPLE. THIS IS A SCAM. THEY DID NOT WARN OF A DEPOSIT FEE AND THEY TOOK 33% OF THE DEPOSIT. I RECOMMEND SUING THEM NOW."*

Another related issue within this category is dubious charges where the user account has been charged, and it is not clear why those charges occur. Abnormally high fees (more than the standard subscription fees) and overcharging of the user account are also captured under this category. For example:

👎 *"It charged me £74.50 when I bought a ticket for £1.50 it's a absolute scam I want my money back!"*

5.2.7 No service. This category mainly covers reviews in which the user complains of not being able to access the app's main functionality after purchase, leading to undesirable consequences for the user. The main difference between the *false advertisement* category and this category is that the former deals with features/functionalities of the app that do not work as advertised. The latter deals with situations where the app does not work at all, i.e., does not even serve its main purpose for the user after the user has made financial commitments in the form of a purchase or subscription. In the example below, the user is fined for illegal parking after paying for parking using the app:

👎 *"Horrible experience with this app. Causing a lot of frustrations with users. when it fails and I get a ticket there is no much help I can get. sometimes I just pay the fines just because the complaint system is awfully inconvenient. I feel cheated and it looks like a money making tool for whoever is collecting the fines."*

Another related example is shown below:

👎 *"I spent 20 euros with all the DLCs included, I feel pretty deceived not being able to play the game."*

5.2.8 Deletion of reviews. This category highlights reviews where the app developers are suspected of deleting reviews left by the user, especially negative reviews. A review captures user feedback, describing their experience of an app, and intending users of an

app typically consult the reviews left by other users on the app distribution platform before downloading the app [48]. Thus, the act of deleting unfavourable reviews by the app developers is perceived as a dishonest practice by the users because leaving only positive reviews may not paint an accurate picture of the app. Users may also feel like the app developers are trying to hide their complaints or other nefarious practices.

It can be argued that certain comments are deleted by app developers because those comments contain ad hominem attacks from the users instead of complaints relating to the app itself. While it is debatable whether app developers are justified in deleting perhaps vitriolic ad hominem comments, we do not make any judgement as to this but simply categorise users' perceptions and complaints of this practice as captured in their reviews. Examples of reviews depicting this accusation are shown below:

- *"I left them a negative review and the developer deleted it. Now I'm going to review them on YouTube and all social media platforms. Basically, they are scammers."*
- *"Deleted my honest review. Warning. Steer clear. They keep trying to make you slip up and pay for premium. I signed up for a free trial last year and they make it too difficult for you to find where to cancel. Was charged about $40... shame such a good app is tarnished by such shady practices."*

5.2.9 Impersonation. An impersonation is an act of pretending to be another person or entity [16]. It also involves the act of giving a false account of the nature of something. This category covers all reviews relating to impersonation or misrepresentation by the app or app developers. This includes scenarios where an app pretends to have the authority of (or relationship to) an organisation when in reality, it has no such relationship. An example review is captured below:

- *"STAY AWAY... this app is a scam. the stickers make it look like it's Brisbane council approved. it's not and they are no help. I still got a fine for using the app correctly and the Brisbane council parking police have no access to check if you have paid or not and do not accept this as a payment method."*

Another example in this category reflects situations where users feel that they are interacting with bots instead of humans when they have signed up to the platform to interact with humans. This is similar to false advertising-related lawsuits of the Match.com platform described in section 1. An example of this is:

- *"Good game, fake players online. I wanted a challenging Monopoly game. But when I start. I can tell that some are bots not real people online. For example, they quickly trade when it is their turn. A normal human will take some time to choose options."*

5.2.10 Fraudulent-looking apps. This category includes reviews reporting suspicious-looking apps based on observations of users or apps deemed to be fake by the users. We created a separate category for these kinds of reviews. Although the users flag the apps in these reviews as fraudulent, they do not provide specific reasons for their accusations beyond their perception of the app as fake or fraudulent. Furthermore, these types of reviews do not fit any of the categories described above, and we sought to highlight them based on the user accusations captured in their reviews. Examples of these reviews include:

- *"...Be careful with this kind of dishonest apps"*
- *"This is a fraud app don't download"*

RQ2 Answer: The result of our analysis of the honesty violations dataset shows that honesty violations can be characterised into ten categories: unfair cancellation and refund policies, false advertisements, delusive subscriptions, cheating systems, inaccurate information, unfair fees, no service, deletion of reviews, impersonation, and fraudulent-looking apps.

6 DISCUSSION AND RECOMMENDATIONS

6.1 Technology (Mobile Apps) as Values Artefacts

Software artefacts such as mobile apps, like other technological artefacts express human values [68]. Although less formally articulated, human values may be reflected throughout the different phases of the software development life cycle [47]. Values are represented in the conception and abstraction of ideas, in the way software features are arranged, described and even implemented and these embodied values are typically those of their creators, e.g., software developers and other stakeholders [31].

Some studies have argued that technological artefacts are value-agnostic tools that can be used for good or bad (i.e., theory of social determination of technology) [27], while others contend that technological artefacts are not value-agnostic, i.e., they hold value qualities and promote certain values over others [70], e.g., the bitcoin blockchain technology [45] is an embodiment of the value category of self-direction. Irrespective of the sociotechnological stance on values in technological (software) artefacts, there is an agreement on the role of software artefacts in changing habits in people and influences society in general, despite the intentions of the software companies behind these artefacts [3, 48]. Sullins writes, "Since the very design capabilities of information technology influence the lives of their users, the moral commitments of the designers of these technologies may dictate the course society will take and our commitments to certain moral values will then be determined by technologists" [66].

Furthermore, while we do not conflate values with ethics (values are the guiding principles of what people consider in life [59] while ethics are the moral expectations that a society agree upon to decide which values are acceptable or not [68]), the value of honesty is an ethically desired value in most societies. Thus we argue for a conscious effort in developing honest software artefacts including mobile apps, and the promotion of honesty in software development practices. Our intention in this paper is not to serve as moral arbiters of values in mobile apps (or other software artefacts) but rather to promote a healthy discussion of these issues in the software research and development community, and point the field towards a critical technical practice of mobile SE, i.e., the reflective work of sociocultural criticisms, highlighting the hidden assumptions in technical processes, and the interaction between the social, cultural and technical aspects of (mobile) SE.

6.2 The Role of App Distribution Platforms

App distribution platforms such as the Apple store and the Google Play store have an important role to play in supporting the values and minimising their violations in apps published on their platforms.

They can serve as enforcers of ethical systems supporting values such as honesty, akin to the manner in which they protect end-users' devices from malicious apps [36, 37]. For instance, they can ensure that app developers are transparent in their billing process and enforce a mandatory multi-step (at least two steps) confirmation not only for subscription but also for in-app purchases.

Another issue on the violation of the value of honesty is related to the non-transparency in the subscription process in apps. For example, while some apps provide a reminder to the user before the end of a trial period so the user can decide to cancel their subscription or progress to a premium service, some other apps provide no reminder whatsoever. A reminder-to-cancel (or upgrade) feature for apps can be necessitated by the distribution platforms to protect the end-user from unintentional subscriptions.

In addition, for games or game-like apps involving the use of money for play, end-users perceptions of unfairness in these systems can be assuaged by a practice of auditing the systems to ensure statistical outcomes that are not only probable but fair to both the end-user and app developers alike, similarly to the way casino systems are routinely audited for fairness and transparency. The results of the audits can then be shown as part of the app information on the app stores.

6.3 Transparent Policies and Agreements

In cases of disputes between end-users and app vendors, where an end-user perceives that they have been unfairly treated, it is typical for the app vendors to refer the end-user to the end-user licence agreement (EULA) signed by the end-user during installation [29]. An EULA is a legally binding contract between the end-user and the app vendor [10].

Some app vendors place their data handling and billing processes in the fine prints of EULAs that are typically difficult to understand by the average user because they are written in legal terms [29]. Some studies have also shown that most end-users who clicked "I agree" do not understand the terms to which they agreed and often expressed genuine concern when the terms are expressed to them [10]. Thus it is important to develop transparent legal policies and easy-to-comprehend EULAs to inform and empower the end-user, and help them understand the terms and implications of these kinds of legal contracts. Transparency and comprehensibility would alleviate wariness and misgivings in this area. Also, we reiterate the position of O'Neill [50], that while transparency may undo secrecy, "it may not limit the deception and deliberate misinformation that undermine relations of trust. If we want to restore trust we need to reduce deception and lies, rather than secrecy" [50]. This area is particularly ripe for interdisciplinary research between the computing sciences, humanities, and law.

6.4 Human Values in SE Research

Research in the broader area of human values in SE is still in its early stages [55]. While the investigation of well-known values such as privacy and security have been considerably developed, other values such as honesty, curiosity, independence have received little attention, possibly due to the subjective and abstract nature of these concepts. This and other recent related works are based on an adaptation of the Schwartz theory of basic human values [60]. However, the nascent field of human values in SE may benefit from new conceptual theories of human values that are more situated closely within SE.

Furthermore, there is the need for the development of tools and techniques, not only in detecting the violation of human values in software artefacts but also providing automatic recommendations for possible fixes. Directions for future work may include the following: the development of approaches for generating end-user comprehensible EULA templates supporting values, approaches for evaluating and auditing fairness in games and game-like systems to support statistically probable results, and modules for static and dynamic analysis tools to detect specific values defects. Another area worth investigating is the development of tools for supporting the inclusion of values throughout the software development lifecycle and the resulting software artefacts including mobile apps.

7 THREATS TO VALIDITY

This section outlines the possible limitations and threats to the validity of our study.

Internal Validity. The qualitative process of building the **honesty_discussion** dataset in Section 3.1 and categorising the different types of honesty violations in Section 5.1 might be biased and error-prone. Hence, it might have introduced some threats to the internal validity of the study. We used three techniques to mitigate such threats. First, the qualitative analysis was conducted iteratively over an ample timeframe to avoid fatigue. Second, each review was analyzed by one analyst and validated by at least one other analyst, followed by several meetings between the analysts to resolve any disagreements and conflicts. Third, the analysts have extensive research experience in the area of human values.

Construct Validity. The analysts might have had different interpretations on the definition of the value of honesty. Our strategy to minimise this threat was making sure the analysts carefully examined seminal papers [60, 61] on the Schwartz theory, formal definition of honesty from dictionaries, and existing software engineering research on human values, including honesty [48, 64]. In this study, among many options, we used five machine learning algorithms to detect honesty violations reviews and four metrics to evaluate the algorithms. Peters et al. [57] claim that it is impracticable to use all algorithms in one study. Hence, we accept that applying other machine learning algorithms to our dataset may lead to different performances. The metrics precision, recall, accuracy, and F1-score used in this study are widely applied and suggested to evaluate machine learning models in software engineering.

External Validity. Our initial sample of app reviews was 236,660 reviews collected from [18] and [48], which was further reduced to 4,885 honesty-related reviews after applying the keywords filter. Our keyword filter may have introduced false negatives and potentially excluded honesty violations in the larger dataset. Hence, we cannot claim that our results are generalisable to all app reviews in the Apple App Store and Google Play Store and other platforms (e.g., online marketplaces).

8 CONCLUSION

Mobile software applications (apps) are very widely used and applied and hence need to reflect critical human value considerations

such as curiosity, freedom, tradition, and honesty. The support for – or violation of – these critical human values in mobile apps have been shown to be captured in app reviews. In this work we focused on the value of honesty. We presented an approach for automatically finding app reviews that reveal the violation of the human value of honesty from an end-user perspective. In developing our automated approach, we evaluated five different algorithms using a manually annotated and validated dataset of app reviews. Our evaluation shows that the Support Vector Machine (SVM) algorithm provides a higher accuracy than the other algorithms in detecting the violation of the value of honesty in app reviews, and also surpasses a baseline random classifier with an F1 score of 0.89. We also characterised the different kinds of honesty violations reflected in app reviews. Our manual qualitative analysis of the reviews containing honesty violations resulted in ten categories: unfair cancellation and refund policies, false advertisements, delusive subscriptions, cheating systems, inaccurate information, unfair fees, no service, deletion of reviews, impersonation, and fraudulent-looking apps. The results of our study highlight the importance of considering software artefacts, such as mobile apps, as an embodiment of human values with consequences on end-users and society as a whole. We emphasise the role of app distribution platforms in supporting human values, such as honesty, on their platforms, and discuss the need for the software engineering research community to investigate methods and tools to better minimise the violation of human values in software artefacts.

ACKNOWLEDGMENTS

This work is supported by ARC Discovery Grant DP200100020. Grundy is supported by ARC Laureate Fellowship FL190100035. Li is supported by ARC DECRA DE200100016.

REFERENCES

[1] 2016. *Cheating, Corruption, and Concealment.* Cambridge University Press, 1–12.
[2] Charu C Aggarwal and ChengXiang Zhai. 2012. A survey of text classification algorithms. In *Mining text data*. Springer, 163–222.
[3] Philip E. Agre. 1997. *Social Science, Technical Systems and Cooperative Work: Beyond the Great Divide.* Erlbaum, Chapter Toward a Critical Technical Practice: Lessons Learned in Trying to Reform AI.
[4] Huib Aldewereld, Virginia Dignum, and Yao-hua Tan. 2015. *Design for ValuesInformation and communication technologies in Software Development.* Springer Netherlands, Dordrecht, 831–845.
[5] Eman Abdullah AlOmar, Wajdi Aljedaani, Murtaza Tamjeed, Mohamed Wiem Mkaouer, and Yasmine N. El-Glaly. 2021. Finding the Needle in a Haystack: On the Automatic Identification of Accessibility User Reviews. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 387, 15 pages. https://doi.org/10.1145/3411764.3445281
[6] Anonymous. 2022. The Replication Repository of this Manuscript:. (2022). https://anonymous.4open.science/r/ml_app_reviews-3ED6/README.md
[7] Emma Bowman. [n. d.]. After Data Breach Exposes 530 Million, Facebook Says It Will Not Notify Users. (2021) https://www.npr.org/2021/04/09/986005820/after-data-breach-exposes-530-million-facebook-says-it-will-not-notify-users.
[8] John L Campbell, Charles Quincy, Jordan Osserman, and Ove K Pedersen. 2013. Coding in-depth semistructured interviews: Problems of unitization and intercoder reliability and agreement. *Sociological Methods & Research* 42, 3 (2013), 294–320.
[9] Laura V. Galvis Carreño and Kristina Winbladh. 2013. Analysis of user comments: An approach for software requirements evolution. In *2013 35th International Conference on Software Engineering (ICSE)*. 582–591. https://doi.org/10.1109/ICSE.2013.6606604
[10] Florence M Chee, Nicholas T Taylor, and Suzanne de Castell. 2012. Re-Mediating Research Ethics: End-User License Agreements in Online Games. *Bulletin of science, technology & society* 32, 6 (2012), 497–506.
[11] An-Shou Cheng and Kenneth R. Fleischmann. 2010. Developing a Meta-Inventory of Human Values. In *Proceedings of the 73rd ASIS&T Annual Meeting on Navigating Streams in an Information Ecosystem*, Vol. 47. American Society for Information Science, Article 3, 10 pages.
[12] Adelina Ciurumelea, Andreas Schaufelbühl, Sebastiano Panichella, and Harald C. Gall. 2017. Analyzing reviews and code of mobile apps for better release planning. In *2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER)*. 91–102. https://doi.org/10.1109/SANER.2017.7884612
[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 4171–4186.
[14] Andrea Di Sorbo, Sebastiano Panichella, Carol V. Alexandru, Junji Shimagaki, Corrado A. Visaggio, Gerardo Canfora, and Harald C. Gall. 2016. What Would Users Change in My App? Summarizing App Reviews for Recommending Software Changes. In *Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering* (Seattle, WA, USA) *(FSE 2016)*. Association for Computing Machinery, New York, NY, USA, 499–510. https://doi.org/10.1145/2950290.2950299
[15] Collins Dictionary. [n. d.]. Definition of 'honesty'. (2021) https://www.collinsdictionary.com/dictionary/english/honesty.
[16] Collins Dictionary. [n. d.]. Definition of 'impersonate'. (2021) https://www.collinsdictionary.com/dictionary/english/impersonate.
[17] Feng Dong, Haoyu Wang, Li Li, Yao Guo, Tegawendé F Bissyandé, Tianming Liu, Guoai Xu, and Jacques Klein. 2018. Frauddroid: Automated ad fraud detection for android apps. In *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 257–268.
[18] Marcelo Medeiros Eler, Leandro Orlandin, and Alberto Dumont Alves Oliveira. 2019. Do Android App Users Care about Accessibility? An Analysis of User Reviews on the Google Play Store. In *Proceedings of the 18th Brazilian Symposium on Human Factors in Computing Systems* (Vitória, Espírito Santo, Brazil) *(IHC '19)*. Association for Computing Machinery, New York, NY, USA, Article 23, 11 pages.
[19] Martin Fochmann, Nadja Fochmann, Martin G. Kocher, and Nadja Müller. 2021. Dishonesty and risk-taking: Compliance decisions of individuals and groups. *Journal of Economic Behavior & Organization* 185 (2021), 250–286. https://doi.org/10.1016/j.jebo.2021.02.018
[20] Barney G Glaser, Anselm L Strauss, and Elizabeth Strutzel. 1968. The discovery of grounded theory; strategies for qualitative research. *Nursing research* 17, 4 (1968), 364.
[21] Don Gotterbarn, Amy Bruckman, Catherine Flick, Keith Miller, and Marty J. Wolf. 2017. ACM Code of Ethics: A Guide for Positive Action. *Commun. ACM* 61, 1 (dec 2017), 121–128. https://doi.org/10.1145/3173016
[22] Emitza Guzman and Walid Maalej. 2014. How Do Users Like This Feature? A Fine Grained Sentiment Analysis of App Reviews. In *2014 IEEE 22nd International Requirements Engineering Conference (RE)*. 153–162. https://doi.org/10.1109/RE.2014.6912257
[23] Yangyu Hu, Haoyu Wang, Yajin Zhou, Yao Guo, Li Li, Bingxuan Luo, and Fangren Xu. 2019. Dating with scambots: Understanding the ecosystem of fraudulent dating applications. *IEEE Transactions on Dependable and Secure Computing* (2019).
[24] Waqar Hussain, Harsha Perera, Jon Whittle, Arif Nurwidyantoro, Rashina Hoda, Rifat Ara Shams, and Gillian Oliver. 2020. Human Values in Software Engineering: Contrasting Case Studies of Practice. *IEEE Transactions on Software Engineering* (2020), 1–15.
[25] Claudia Iacob and Rachel Harrison. 2013. Retrieving and analyzing mobile apps feature requests from online reviews. In *2013 10th Working Conference on Mining Software Repositories (MSR)*. 41–44. https://doi.org/10.1109/MSR.2013.6624001
[26] Nicolas Jacquemet, Alexander G James, Stéphane Luchini, James J Murphy, and Jason F Shogren. 2021. Do truth-telling oaths improve honesty in crowd-working? *PloS one* 16, 1 (2021), 1–18.
[27] Bernward Joerges. 1999. Do Politics Have Artefacts? *Social Studies of Science* 29, 3 (1999), 411–431. https://doi.org/10.1177/030631299029003004
[28] Ralph Keyes. 2004. *The post-truth era : dishonesty and deception in contemporary life* (1st ed. ed.). St. Martin's Press, New York.
[29] Chelsea King. 2017. Forcing players to walk the plank: why end user license agreements improperly control players' rights regarding microtransactions in video games. *William and Mary law review* 58, 4 (2017), 1365.
[30] J.M. Lang. 2013. *Cheating Lessons: Learning from Academic Dishonesty.* Harvard University Press. https://books.google.fm/books?id=hTeImwEACAAJ
[31] John Lennox. 2020. *2084: Artificial Intelligence, the Future of Humanity, and the God Question.* Zondervan.
[32] Stanislav Levin and Amiram Yehudai. 2017. Boosting Automatic Commit Classification Into Maintenance Activities By Utilizing Source Code Changes. In *Proceedings of the 13th International Conference on Predictive Models and Data Analytics in Software Engineering* (Toronto, Canada) *(PROMISE)*. Association for Computing Machinery, New York, NY, USA, 97–106.
[33] Stanislav Levin and Amiram Yehudai. 2019. Towards Software Analytics: Modeling Maintenance Activities. *CoRR* abs/1903.04909 (2019). arXiv:1903.04909

http://arxiv.org/abs/1903.04909

[34] Conghui Li, Humphrey O. Obie, and Hourieh Khalajzadeh. 2021. A First Step Towards Detecting Values-violating Defects in Android APIs. arXiv:2109.14359 [cs.SE]

[35] Huiying Li, Li Zhang, Lin Zhang, and Jufang Shen. 2010. A user satisfaction analysis approach for software evolution. In *2010 IEEE International Conference on Progress in Informatics and Computing*, Vol. 2. 1093–1097. https://doi.org/10.1109/PIC.2010.5687999

[36] Li Li, Kevin Allix, Daoyuan Li, Alexandre Bartel, Tegawendé F Bissyandé, and Jacques Klein. 2015. Potential component leaks in Android apps: An investigation into a new feature set for malware detection. In *2015 IEEE International Conference on Software Quality, Reliability and Security*. IEEE, 195–200.

[37] Li Li, Daoyuan Li, Tegawendé F Bissyandé, Jacques Klein, Haipeng Cai, David Lo, and Yves Le Traon. 2017. Automatically locating malicious packages in piggybacked android apps. In *2017 IEEE/ACM 4th International Conference on Mobile Software Engineering and Systems (MOBILESoft)*. IEEE, 170–174.

[38] Xiaozhou Li, Zheying Zhang, and Kostas Stefanidis. 2018. Mobile App Evolution Analysis Based on User Reviews. In *SoMeT*.

[39] Everton da Silva Maldonado, Emad Shihab, and Nikolaos Tsantalis. 2017. Using Natural Language Processing to Automatically Detect Self-Admitted Technical Debt. *IEEE Transactions on Software Engineering* 43, 11 (2017), 1044–1062. https://doi.org/10.1109/TSE.2017.2654244

[40] Nina Mazar and Dan Ariely. 2006. Dishonesty in Everyday Life and Its Policy Implications. *Journal of Public Policy & Marketing* 25, 1 (2006), 117–126. http://www.jstor.org/stable/30000530

[41] Christian B Miller. 2021. *Honesty: The Philosophy and Psychology of a Neglected Virtue*. Oxford University Press USA - OSO, Oxford.

[42] Elizabeth R Morrissey. 1974. Sources of error in the coding of questionnaire data. *Sociological Methods & Research* 3, 2 (1974), 209–232.

[43] Davoud Mougouei. 2020. Engineering Human Values in Software through Value Programming. *Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops* (2020), 133–136.

[44] Davoud Mougouei, Harsha Perera, Waqar Hussain, Rifat Shams, and Jon Whittle. 2018. Operationalizing Human Values in Software: A Research Roadmap *(ESEC/FSE 2018)*. Association for Computing Machinery, New York, NY, USA, 780–784. https://doi.org/10.1145/3236024.3264843

[45] Satoshi Nakamoto. 2009. Bitcoin: A peer-to-peer electronic cash system. http://www.bitcoin.org/bitcoin.pdf

[46] William S Noble. 2006. What is a support vector machine? *Nature biotechnology* 24, 12 (2006), 1565–1567.

[47] Arif Nurwidyantoro, Mojtaba Shahin, Michel Chaudron, Waqar Hussain, Harsha Perera, Rifat Ara Shams, and Jon Whittle. 2021. Towards a Human Values Dashboard for Software Development: An Exploratory Study. In *Proceedings of the 15th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)* (Bari, Italy) *(ESEM '21)*. Association for Computing Machinery, New York, NY, USA, Article 23, 12 pages.

[48] Humphrey O. Obie, Waqar Hussain, Xin. Xia, John Grundy, Li Li, Burak Turhan, Jon Whittle, and Mojtaba Shahin. 2021. A First Look at Human Values-Violation in App Reviews. In *2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS)*. 29–38.

[49] Humphrey O. Obie, Mojtaba Shahin, John Grundy, Burak Turhan, Li Li, Waqar Hussain, and Jon Whittle. 2021. Does Domain Change the Opinion of Individuals on Human Values? A Preliminary Investigation on eHealth Apps End-users. arXiv:2110.01832 [cs.SE]

[50] Onora O'Neill. [n. d.]. Trust is the first casualty of the cult of transparency. (2002) https://www.telegraph.co.uk/comment/personal-view/3575750/Trust-is-the-first-casualty-of-the-cult-of-transparency.html.

[51] Fabio Palomba, Mario Linares-Vásquez, Gabriele Bavota, Rocco Oliveto, Massimiliano Di Penta, Denys Poshyvanyk, and Andrea De Lucia. 2015. User reviews matter! Tracking crowdsourced reviews to support evolution of successful apps. In *2015 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. 291–300. https://doi.org/10.1109/ICSM.2015.7332475

[52] Sebastiano Panichella, Andrea Di Sorbo, Emitza Guzman, Corrado A. Visaggio, Gerardo Canfora, and Harald C. Gall. 2015. 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)*. 281–290. https://doi.org/10.1109/ICSM.2015.7332474

[53] Lucas Pelloni, Giovanni Grano, Adelina Ciurumelea, Sebastiano Panichella, Fabio Palomba, and Harald C. Gall. 2018. BECLoMA: Augmenting stack traces with user review information. In *2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER)*. 522–526. https://doi.org/10.1109/SANER.2018.8330252

[54] Harsha Perera, Waqar Hussain, Davoud Mougouei, Rifat A. Shams, Arif Nurwidyantoro, and Jon Whittle. 2019. Towards Integrating Human Values into Software: Mapping Principles and Rights of GDPR to Values. In *2019 IEEE 27th International Requirements Engineering Conference (RE)*. 404–409.

[55] Harsha Perera, Waqar Hussain, Jon Whittle, Arif Nurwidyantoro, Davoud Mougouei, Rifat Ara Shams, and Gillian Oliver. 2020. A Study on the Prevalence of Human Values in Software Engineering Publications, 2015 – 2018. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering* (Seoul, South Korea) *(ICSE '20)*. Association for Computing Machinery, New York, NY, USA, 409–420. https://doi.org/10.1145/3377811.3380393

[56] Sarah Perez. [n. d.]. Dating app maker Match sued by FTC for fraud. (2019) https://techcrunch.com/2019/09/26/dating-app-maker-match-sued-by-ftc-for-fraud/.

[57] Fayola Peters, Thein Than Tun, Yijun Yu, and Bashar Nuseibeh. 2017. Text filtering and ranking for security bug report prediction. *IEEE Transactions on Software Engineering* 45, 6 (2017), 615–631.

[58] Minh Vu Phong, Tam The Nguyen, Hung Viet Pham, and Tung Thanh Nguyen. 2015. Mining User Opinions in Mobile App Reviews: A Keyword-Based Approach. In *2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. 749–759. https://doi.org/10.1109/ASE.2015.85

[59] Milton Rokeach. 1973. *The Nature of Human Values*. Free Press.

[60] Shalom Schwartz. 1992. Universals in the Content and Structure of Values: Theoretical Advances and Empirical Tests in 20 Countries. *Advances in Experimental Social Psychology* 25 (1992).

[61] Shalom Schwartz. 2012. An Overview of the Schwartz Theory of Basic Values. *Online Readings in Psychology and Culture* 2 (2012).

[62] Norbert Seyff, Florian Graf, and Neil Maiden. 2010. Using Mobile RE Tools to Give End-Users Their Own Voice. In *2010 18th IEEE International Requirements Engineering Conference*. 37–46. https://doi.org/10.1109/RE.2010.15

[63] Poole Shaffery. [n. d.]. CYBER SECURITY: When the Cover Up Is Worse Than the Crime: Uber & the Consequences of Hiding a Data Breach. (2021) https://www.pooleshaffery.com/news/2017/december/cyber-security-when-the-cover-up-is-worse-than-t/.

[64] Rifat Ara Shams, Waqar Hussain, Gillian Oliver, Arif Nurwidyantoro, Harsha Perera, and Jon Whittle. 2020. Society-oriented applications development: Investigating users' values from bangladeshi agriculture mobile applications. In *2020 IEEE/ACM 42nd International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS)*. IEEE, 53–62.

[65] Rifat Ara Shams, Mojtaba Shahin, Gillian Oliver, Waqar Hussain, Harsha Perera, Arif Nurwidyantoro, and Jon Whittle. 2021. Measuring Bangladeshi Female Farmers' Values for Agriculture Mobile Applications Development. In *54th Hawaii International Conference on System Sciences, HICSS'21*. 1–10.

[66] John Sullins. [n. d.]. Information Technology and Moral Values. (2018) https://plato.stanford.edu/entries/it-moral-values/.

[67] Afua van Haasteren, Felix Gille, Marta Fadda, and Effy Vayena. 2019. Development of the mHealth App Trustworthiness checklist. *Digital health* 5 (2019), 2055207619886463.

[68] Jon Whittle. 2019. Is Your Software Valueless? *IEEE Software* 36, 3 (2019), 112–115. https://doi.org/10.1109/MS.2019.2897397

[69] Jon Whittle, Maria Angela Ferrario, Will Simm, and Waqar Hussain. 2021. A Case for Human Values in Software Engineering. *IEEE Software* 38, 1 (2021), 106–113. https://doi.org/10.1109/MS.2019.2956701

[70] Langdon Winner. 1980. Do Artifacts Have Politics? *Daedalus* 109, 1 (1980), 121–136. http://www.jstor.org/stable/20024652

[71] Emily Winter, Steve Forshaw, and Maria Angela Ferrario. 2018. Measuring Human Values in Software Engineering. In *2018 ACM/IEEE 12th International Symposium on Empirical Software Engineering and Measurement*. 1–4.

[72] Roulla Yiacoumi. [n. d.]. Online educator Shaw Academy to refund students: 'Free trial' charged students even when they cancelled. (2021) https://ia.acs.org.au/article/2021/online-educator-shaw-academy-to-refund-students.html.

[73] Liming Zhu, Xiwei Xu, Qinghua Lu, Guido Governatori, and Jon Whittle. 2021. AI and Ethics – Operationalising Responsible AI. arXiv:2105.08867 [cs.AI]

[74] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*. 19–27.