Automated Identification of Sexual Orientation and Gender Identity Discriminatory Texts from Issue Comments

SAYMA SULTANA, Wayne State University, USA
JAYDEB SARKER, University of Nebraska Omaha, USA
FARZANA ISRAT, Wayne State University, USA
RAJSHAKHAR PAUL, Idaho State University, USA
AMIANGSHU BOSU, Wayne State University, USA

In an industry dominated by straight men, many developers representing other gender identities and sexual orientations often encounter hateful or discriminatory messages. Such communications pose barriers to participation for women and LGBTQ+ persons. Due to sheer volume, manual inspection of all communications for discriminatory communication is infeasible for a large-scale Free Open-Source Software (FLOSS) community. To address this challenge, this study proposes an automated mechanism to identify Sexual Orientation and Gender Identity Discriminatory (SGID) texts in software developers' communications. On this goal, we trained and evaluated SGID4SE (Sexual orientation and Gender Identity Discriminatory text identification for (4) Software Engineering texts), a supervised learning-based tool. SGID4SE incorporates six preprocessing steps and ten state-of-the-art algorithms. SGID4SE employs six distinct strategies to enhance the performance of the minority class. We empirically evaluated each strategy and identified an optimum configuration for each algorithm. In our ten-fold cross-validation-based evaluations, a BERT-based model achieves the best performance with 85.9% precision, 80.0% recall, and 82.9% F1-score for the SGID class. This model achieves 95.7% accuracy and a Matthews Correlation Coefficient of 80.4%. Our dataset and tool establish a foundation for further research in this direction.

CCS Concepts: • **Software and its engineering** → **Collaboration in software development**; **Integrated and visual development environments**; • **Computing methodologies** → **Supervised learning**.

Additional Key Words and Phrases: misogyny, sexism, discrimination, hate speech, pejorative

ACM Reference Format:
Sayma Sultana, Jaydeb Sarker, Farzana Israt, Rajshakhar Paul, and Amiangshu Bosu. 2025. Automated Identification of Sexual Orientation and Gender Identity Discriminatory Texts from Issue Comments. *ACM Trans. Softw. Eng. Methodol.* 34, 0, Article 0 (2025), 31 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

Warning: This paper contains examples of language that some people may find offensive or upsetting.

1 Introduction

According to the 2023 Stack Overflow developer survey [61], only 5.1% of the professional developers around the world identify as women compared to 91.8% identifying as men. In an industry dominated by straight men, many software developers harbor sexist, misogynist, and anti-LGBTQ+ beliefs and

Authors' Contact Information: Sayma Sultana, sayma@wayne.edu, Wayne State University, Detroit, Michigan, USA; Jaydeb Sarker, jsarker@unomaha.edu, University of Nebraska Omaha, Omaha, Nebraska, USA; Farzana Israt, farzanaisrat@wayne.edu, Wayne State University, Detroit, Michigan, USA; Rajshakhar Paul, rpaul@wayne.edu, Idaho State University, Pocatello, Idaho, USA; Amiangshu Bosu, amiangshu.bosu@wayne.edu, Wayne State University, Detroit, Michigan, USA.

ACM 1557-7392/2025/0-ART0
https://doi.org/10.1145/nnnnnnn.nnnnnnn

attitudes, which may vary from subtle to highly overt. Prior research found a presence of sexism and misogyny among various computing organizations [42, 86, 89]. For example, Polly, a software engineer from the United Kingdom, shared the worst feedback she received in a code review: *"I don't care, I only hired you because you wore a skirt in your interview"* [27]. A 2015 survey titled "Elephant in the Valley" shows 84% of women working in Silicon Valley had been called "too aggressive" by their men colleagues [87]. Due to widespread sexist/misogynistic culture and biases against women, 45% women in computing switch careers within ten years, and that attrition rate is more than twice as high for women than it is for men [12]. Discriminating attitudes towards LGBTQ+ persons are also common, as they often are victims of disparaging comments or bullying [35, 81]. These issues are causing attrition of valuable human resources from the computing industry, despite these jobs being in high demand.

Sexual orientation and **G**ender **I**dentity based **D**iscrimination (SGID) have been found among many Free and Libre Open source (FLOSS) communities [64, 78], as women often encounter colleagues who perceive them as technologically less proficient, are assigned menial tasks, and are subject to sexist/misogynistic insults [26, 87]. While women are harassed for mistakes or lack of knowledge, their male colleagues get encouragement for learning from mistakes [77]. As one woman shared her experience, *"Oh, she's a woman. She doesn't know how to code. That's why she did something wrong."* [77]. Not only women but also people identifying as LGBTQ+ encounter negative experiences as their identities are used in derogatory ways. For example, a developer criticized another person's project as *"This is too gay to be true. I'm sorry, this is way too gay, plz delete"*. Discriminatory comments such as these not only demotivate the participation of women and LGBTQ+ persons but also negatively influence efforts to promote diversity, equity, and inclusion (DEI) among FLOSS projects. Therefore, to promote inclusive computing organizations, it is crucial to combat such SGID comments. Although many FLOSS organizations have Codes of Conduct (CoC) to discourage such interactions, these are rarely enforced, as victims are often afraid to report CoC violations due to fears of repercussions [10]. Furthermore, manually checking all interactions is infeasible for project administrators, as large-scale FLOSS communities, such as OpenStack, Wikimedia, Qt, and Apache, regularly generate enormous amounts of text-based communications through various mediums, including code reviews, issue discussions, code commits, and mailing lists. An automated mechanism to flag SGID interactions can assist in two ways. First, it can help project administrators intervene and possibly remove such content. Second, it can also educate people who may not realize that their jokes or remarks are insulting, making many minority groups feel unwelcome. Therefore, the primary objective of this study is *to develop an automated mechanism to identify Sexual orientation and Gender identity Discriminatory (SGID) texts from software developers' communications.*

Although several recent studies have focused on automated identification of sexist and misogynistic communications, those are limited to Twitter posts [43, 45, 69], Reddit discussions [29, 78], and YouTube comments [16]. However, no prior studies have focused on identifying such texts from software developers' communications. A customized SGID tool for the SE is necessary for two reasons. First, some texts may not be considered sexist in a non-SE context. For example, *"Documentation! Is there any lady to add documentation?"* – without knowing that writing software documentation is considered by many as a menial task, a non-SE classifier is less likely to predict this text as sexist. Second, as prior studies have shown the poor performance of off-the-shelf natural language processing (NLP) tools on Software Engineering (SE) communications [47, 71], off-the-shelf tools are unlikely to achieve reliable performances on a SE dataset. Unfortunately, to the best of our knowledge, no such SE domain-specific SGID identification tool or labeled dataset currently exists. To fill this gap, we developed a rubric by conducting a systematic literature review on prior studies that aimed to identify misogynistic texts [82]. This study has been published in ESEM 2021. Subsequently, we modified the rubric to cover derogatory texts toward women and LGBTQ+ people. There are a total of 13 categories, of which 12 belong to the SGID group. To address dataset unbalancing, we adopted a keyword-based selection method used in building prior NLP datasets [11, 51, 88]. Specifically, we employed a systematic approach to curate a set of 252 keywords organized into 12 categories. After searching the GHTorrent export [37] and GitHub search API, we identified 225,117 unique pull request comments, including these keywords. After excluding non-English comments using fastText [17], we were left with a total of 193,056 comments. As this dataset was still too large for manual labeling, we employed a stratified sampling strategy, as used in prior NLP studies [71, 72], to identify 11,007 comments. Each of the selected comments was independently labeled and categorized by two raters with 'substantial agreement' in binary (Cohen's κ [24] =0.658) and an 'acceptable agreement' in multiclass categorization (Krippendorff's α [50] =0.691). We resolved conflicting labels through mutual discussions. After this step, we identified 1,422 (≈13.6%) SGID comments belonging to one of the 12 SGID categories.

Using this dataset, we trained and evaluated SGID4SE (Sexual orientation and Gender Identity based Discrimination identification for (4) Software Engineering texts), as a supervised learning based SGID detection tool. SGID4SE incorporates six preprocessing steps and ten state-of-the-art algorithms. We empirically evaluated each strategy and identified an optimum configuration for each algorithm. In our ten-fold cross-validation-based evaluations, a transformer-based model using the BERT-base encoding [25] boosts the performance with 85.9% precision, 80.0% recall, and 82.9% F1-score for the SGID class. This model achieves 95.7% accuracy and 80.4% Matthews Correlation Coefficient [21]. Our post hoc analyses also identify several lessons that can be useful to develop future SE domain-specific SGID tools. The primary contributions of this research include:

- A classification rubric to manually label SGID texts.
- The first labeled SGID dataset from the SE domain.
- SGID4SE, an automated SGID detection tool for the SE domain.
- Empirical evaluation of optimum configuration for each of the ten algorithms.
- A baseline to improve on and a set of lessons for developing future SE domain-specific SGID tools.
- We release SGID4SE, a labeled dataset, and evaluation results in the replication package on GitHub at: https://github.com/WSU-SEAL/SGID4SE

Notes: A subset of the authors of this paper previously introduced a rubric for identifying sexist and misogynistic content at ESEM 2021 [81]. That work focused on categorizing misogynistic remarks, sexist jokes, and speech-based sexist or misogynistic content. While our current study builds upon the rubric proposed in the ESEM 2021 paper, it differs in two key ways. First, we have refined and extended the original rubric by incorporating insights from additional research on the identification of misogynistic content. The details of our rubric development process are outlined in Section 3. Second, we have implemented an automated tool for detecting such content following the rubric. A preliminary version of this tool was presented in the Student Research Competition at ASE 2022 [80]. Upon closer examination of the results and datasets, we can find significant differences between these two papers. The SRC paper presents initial findings based on a smaller dataset without any fine-tuning. For instance, the best-performing model in that paper reports 80% precision, 67.07% recall, 72.5% F1 score, and 95.96% accuracy. In contrast, the current tool enhances performance through several strategies, including adding LGBTQ+ related keywords and samples, pipeline optimization, techniques for addressing dataset imbalance, and error analysis for the best model. As a result, this study reports an improved performance with 85.9% precision, 80.0% recall, and 82.9% F1 score for the SGID class, achieving an accuracy of 95.7% and a Matthews Correlation Coefficient of 80.4%.

Organization: The remainder of the paper is organized as follows. Section 2 discusses related works. Section 3 describes our methodology for developing a labeled SGID dataset. Section 4 details the design of SGID4SE. Sections 5 and 6 present evaluation results and discuss the implications of this study, respectively. Section 7 addresses the limitations, while Section 8 concludes the paper.

2 Related Works

Anti-social behavior among software projects. Numerous research have demonstrated the presence of toxic content in FLOSS communication channels like IRC chat and mailing lists [31, 78]. Developers working on FLOSS projects have reported insults, attacks, and other forms of toxicity [67]. Miller *et* al. [56] also observed a unique form of toxicity in FLOSS projects that differ from those in other platforms like Reddit or Wikipedia. Such toxic communication comprising entitlements, insult, and arrogance, leads to tension and exhaustion for the developers, and is "likely to make someone leave" [56, 67]. Ferreira *et* al. [31] also looked at incivility among contributors in FLOSS projects and discovered that incivility can take many different forms, including bitter frustration, name-calling, mockery, and threats. "Pushback", a phenomenon where a reviewer blocks the modification request due to a personal conflict, is the result of such uncivil behavior. The Google Jigsaw AI team developed a guidebook [9] for identifying toxic content and the Google perspective API [8] for the general domain for automatic toxicity detection. Sarker *et* al. [71] showed that toxicity detector tools developed for the general domain do not perform well for the particular domain of software developers. The need for automatic toxicity detector tools for software developers prompts the creation of STRUDEL [67] and ToxiCR [72].

Research on sexism and misogyny identification. Online misogyny in different platforms, e.g., Twitter [22, 43, 45, 90], Reddit [40], and YouTube [16] have been subjected to research by quite a few researchers. Automated identification of sexist and misogynistic texts can build a healthy environment for women so that they can participate and share their thoughts. On this goal, the Automatic Misogyny Identification (AMI) task hosted at the 2018 IberEval released two labeled datasets of English and Spanish tweets [33]. They also provided a classification rubric for five types of misogynistic texts: i) Stereotype & objectification, ii) Dominance, iii) Derailing, iv) Sexual harassment & threats of Violence, and v) Discredit. This competition resulted in the development and evaluation of several AMI approaches [6, 11, 19, 34, 36, 54, 60, 63, 76]. On the one hand, while the tool proposed by Pamungkas *et* al. [62] achieved the best accuracy of 91%, it achieved only 36.9% f1-score in identifying misogynous English texts. On the other hand, the best f1-score of 79% was achieved by Shushkevich *et* al. [76] using an ensemble of Naïve Bayes(NB) and Support Vector Machine(SVM) classifiers with an accuracy of 70.6%. Motivated by the IberEval, EVALITA released a labeled AMI dataset of Italian tweets and hosted a competition to develop tools using that dataset [33]. This competition was repeated in 2020 using a new dataset [14] where Muti *et* al. [59] achieved the best f1-score of 74.3% using AlBERTo [66], i.e., a pre-trained BERT model for Italian. While datasets and tools for investigating misogyny in the general domain are available and have been explored, misogyny in FLOSS projects has not been studied yet. While investigating profanity and insults in FLOSS projects, Squire *et* al. [78] found three types of gender-based insults: maternal insult, sexual double entendre jokes, and the use of women relatives to represent unintelligent persons. However, no prior study focused on studying gender-based insults or derogatory content for the specific domain of software developers.

Relationship between toxicity/incivility and SGID. Toxicity encompasses a wide range of negative behaviors, with gender discrimination being just one specific type. The Perspective API, developed by Jigsaw and Google, characterizes toxicity as "A rude, disrespectful, or unreasonable comment that is likely to make you leave a discussion"[1]. Importantly, it does not explicitly address gender issues. As outlined in Sarker *et* al.'s rulebook [72], flirtation and identity attacks are included in toxic content. Based on their rules, identity attacks based on gender and sexual orientation and flirtation fall under the definition of toxicity. For example, *"Why you gay? Why you gay? hmm" or "hey pretty girl"* can be identified as toxic and SGID content. In contrast to Sarker *et* al.'s rulebook, Miller *et* al. [56] define toxicity as "An umbrella term for various antisocial behaviors including trolling, flaming, hate speech, harassment, arrogance, entitlement, and cyberbullying." Here, the intersection with SGID and toxicity is primarily in the context of harassment related to gender or sexual orientation. Therefore, SGID contents represent a specific subset of toxicity/incivility. However, not all forms of SGID content are adequately addressed by the toxicity/incivility definitions of Sarker *et* al. [72], Miller *et* al. [56], and Ferreira *et* al [31]. For example, prior research has identified stereotyping as a type of misogynistic and SGID content [11, 81]. However, this type of content would not be flagged as toxic according to the standard toxicity definition. For example, *"What does a blonde do when her computer freezes.......she sticks it in the microwave :P* - this comment expresses stereotyping about women which will not be identified as toxic. But such type of comment should be identified as SGID content. Therefore, if a generic toxicity detection classifier is used, it may not effectively identify all instances of SGID content in communication excerpts.

3 Dataset to Train and Evaluate Automated SGID Identification Models

We created a large-scale manually labeled SGID dataset with a five-step methodology, as follows: 1) defining SGID, 2) developing a rubric for manual labeling, 3) employing a sampling strategy to create a dataset of SE communications with a higher ratio of SGID texts than randomly selected ones, 4) labeling this dataset using multiple human raters, and 5) retraining existing tools on our dataset. We detail these five steps in the following subsections.

3.1 Step 1: Research Context Definition

Richard Schaefer states, "Sexism may be defined as an ideology based on the belief that one sex is superior to another" [74]. This ideology points to biological differences to claim superiority and justify men's dominance over women [84]. While persons from all genders may be the object of sexist attitudes, women have been the usual victims [75]. On the other hand, "misogyny" derives from the Ancient Greek word "mīsoguníā", which means hatred towards women [79]. Misogyny is often expressed in terms of male dominance, sexual harassment, belittling of women, intimidation, violence against women, and sexual objectification [23, 44, 48]. While earlier literature on sexism and misogyny primarily focus on the binary genders (men vs. women), prejudice or discrimination towards LGBTQ+[2] persons are not uncommon among software developers [35, 81]. To include LGBTQ+ persons, this study defines SGID as an umbrella encompassing sexist, misogynistic, and anti-LGBTQ+ expressions. Therefore:

> *"A text is considered as sexual orientation and gender identity discriminatory (SGID), if it expresses prejudice or discrimination based on a person's gender, biological sex, gender identity, or sexual orientation."*

Based on this definition, a straight man can also be a target of SGID. However, our target automated model focuses on SGIDs against minorities (i.e., women and LGBTQ+) since these groups are more likely to be marginalized due to SGIDs.

[1] https://developers.perspectiveapi.com/

[2] According to the youth.gov guidelines (https://youth.gov/youth-topics/lgbt), LGBTQ+ is an umbrella term that encompasses various sexual orientations such as lesbian, gay, bisexual, transgender, queer or questioning, as well as other diverse gender identities.

Table 1. Rubric to label Gender identity and Sexual orientation Discriminatory (SGID) texts with examples from our dataset

Inclusion criteria	Example	Category
Implies random mean speech to demean women/LGBTQ+ people without any particular intention	*Bimbos can't get them, is this intentional or a bug?*	Discredit
Derogates women or LGBTQ+ people based on traditional gender roles, personal characteristics, physical or sexual limitations and to make them feel subordinated	*What does a blonde do when her computer freezes.......she sticks it in the microwave :P*	Stereotyping
Refers to or jokes about one's sexuality or sexual orientation in an aggressive manner	*Why you gay? Why you gay? hmm*	Sexual harassment
Intends to assert power over women or LGBTQ+ persons through threats to intimidate or silence them	"BITCH I WILL KILL YOU"	Threats of violence
Intends to control women or LGBTQ+ people to exclude them from the conversation	*.. I'd love to be able to mute it when the drama queens start using Discord as a soapbox to bypass /ignore.*	Dominance
Blames the person who encounters problems for aggression generated by others	"She is too sensitive, she did not get the joke about women"	Victim blaming
Refers to women or LGBTQ+ people as objects by creating sexual imagery of body parts or themselves	*Would you like to get some tickets? With horny girls?*	Sexual objectification
Discusses women's or LGBTQ+ people's physical appearance or clothes	*Please Update README.md Blondie is my girl*	Appearance reference
Insults or jokes directed towards a person's women relatives	*This is cheating harder than your mom does.*	Maternal insults
Expresses ill wish or hatred towards women or LGBTQ+ persons	*I hate gays, cuz they are really gay*	Damning
Demeans or insults LGBTQ+ persons or groups using LGBTQ+ slurs	*A faggot wrote the source code.*	Anti-LGBTQ+
Not directed to women or LGBTQ+ people but mentions uncomfortable references about gender or sex	*I'm having trouble finding the balls! I know how to ligma but not how to suck*	Sexual references
Fits none of the categories	If *I'd only read the whole source that I cited. Grammar Girl agrees with you. Apologies. Leave it as-is.*	Non-SGID

3.2 Step 2: Classification Rubric

In the first phase, we focused on developing a binary labeling rubric (i.e., whether a comment is SGID or not). We took the guidelines provided by Guest *et* al. [40] as our starting point. They provided several inclusion and exclusion criteria to determine when a text should be labeled as misogynous or non-misogynous. Our adoption of their inclusion criteria includes i) adding prejudice or discrimination against LGBTQ+ persons, ii) adding LGBTQ+ slurs in the list of derogatory terms, and iii) categorizing comments with misogynist or hateful pejoratives towards LGBTQ+ people as SGID.

Although we adopted all of their inclusion criteria by broadening scope, we modified one of their exclusion criteria for two primary reasons. First, while Guest *et* al.'s rubric focuses on identifying misogynous texts, our's focuses on a broader umbrella, including sexism and anti-LGBTQ+ expressions. Second, they did not consider object-directed misogynous pejoratives such as "Git is a bitch" as misogynous. As our rubric is targeted toward professional workplace communication in contrast to theirs being targeted toward everyday Twitter posts, we consider object-directed pejoratives as SGIDs. While some readers may disagree with this change, prior research has shown that words such as 'bitch, and 'cunt' has strong misogynistic stereotyping roots, and calling someone or some object 'bitch' indicates that the target is not conforming to the caller's expected standards similar to a "malicious, spiteful, or overbearing woman" [30]. Table 1 lists our 12 inclusion criteria with appropriate examples and corresponding categories for SGID texts.

After developing a binary classification rubric for SGID vs. non-SGID comments, we focused on developing an SGID classification scheme to record what type of SGID texts occur more frequently on GitHub. On this goal, we adopted Sultana *et* al.'s sexist and misogynistic text classification rubric developed for the SE domain [83]. They proposed three classification schemes for sexist, misogynistic and jokes targeting women. We merged their three schemes into one common umbrella. We found some categories with different names across these three schemas. We merged such categories into one. For example, we merged one subcategory: 'Derailing: reject male responsibility, and attempt to disrupt the conversation to refocus it' with 'Victim blaming: blaming the victims for the problems they face' since both express similar notions of misogyny. We also excluded the 'mixed bias: gender bias mixed with other types of bias, e.g., religious or regional bias' subcategory since we want to focus only on gender and sexual orientation-related biases. Although Sultana *et* al. did not include a subcategory for maternal insults, we included this subcategory as prior studies have shown texts in IRC chats and emails that involve *'Mom jokes'* [78] or sexist jokes related to woman relatives such as *Grandma test*, *Aunt Tillie test*, and *Girlfriend tests* [5]. At this step, we could map 10 out of our 12 inclusion criteria, each to a different SGID category. To map the remaining two inclusion criteria, we created two additional categories. The Anti-LGBTQ+ category records prejudice and discrimination against LGBTQ+ persons, and the 'Sexual reference' category records flirtations and references to sexual activities that may be uncomfortable to persons of gender identities. The last column in Table 1 lists our mapping from inclusion criteria to SGID categories. We also include an example for each category taken from our dataset. Among the thirteen categories, twelve belong to the SGID group, and the remaining one forms the Non-SGID group, which includes neutral texts and other inappropriate content that are not discriminatory based on gender or sexual orientation. However, non-SGID texts may still be toxic, racist, or inappropriate for other reasons.

3.3 Step 3: Dataset Creation

Although SGID texts exist in software developers' communications, they are not frequent. Even in the general domain, such as Twitter or YouTube comments, a random selection would find a negligible ratio of SGID texts [90]. To overcome such challenges, prior studies selected Tweets based on certain predefined keywords [11] or hashtags [69], selected texts from specific Reddit channels [78], as well as women-oriented blogs and forums [16]. Motivated by those examples [11, 51, 88], we adopted a keyword-based sampling.

3.3.1 Keyword Selection. By analyzing existing SGID datasets and their development methodologies [11, 78, 88, 90], we established eleven distinct categories of discussion areas (i.e., the topic that a sentence or paragraph focuses on), all of which can indicate the presence of SGID comments. The first column of Table 3 lists those categories, and the second column provides a brief rationale for why a category may appear in SGID contexts. After formulating our keyword categories, we looked into prior studies and online lists (listed in Table 2) on sexism, misogyny, hate speech, pejoratives, and LGBTQ+ terms to identify possible keywords for our categories. We manually inspected each list to identify words fitting one of the eleven discussion areas listed in Table 3. We would like to point out that a direct mapping between these discussion areas and our SGID categories listed in

Table 2. Keyword sources and number of words taken from each source. A word may belong to multiple lists.

Source	Keyword type	Rationale	#
Hewitt *et* al. [43]	Misogyny context	Authors has listed commonly found misogynistic keywords to study online misogyny	17
SemEval 2019 [57]	Hate speech against women	In this dataset organizers listed hateful words against women	21
Guest *et* al. [40]	Pejorative terms for women	These terms are explicitly insulting and derogatory, like "slut" or "whore," or implicitly convey negativity or hostility toward women, such as "Stacy" or "Becky."	15
Baucom, Erin [15]	LGBTQ+ terms	Authors listed keywords that has been used	14
Kurita *et* al. [52]	Hatred and identity attack	This list contains words related to women body parts and misogynistic pejorative	100
Hatebase [1]	Toxic and swear words	This is the world's largest structured repository of regionalized, multilingual hate speech and used in prior study [29] to identify misogyny in online	54
List of gendered nouns [85]	Gender specific roles	We have taken women-specific gendered nouns and roles that might be used to demean or express stereotypes about women	11
Wikipedia	Pejorative terms for women [3]	This is the list keywords to belittle or derogate women	19
Wikipedia	LGBTQ+ terms [2]	It contains the list of words that are used to refer LGBTQ+ people	24

Table 1 is not possible, as some of the areas, such as women's roles, women relatives, or general women specific words can appear at various SGID contexts. For example, 'I hate girls' belongs to 'Damning' but 'Let's go to score some horny girls' belongs to 'Sexual objectification'. Table 2 also lists the number of keywords taken from each source based on our manual inspections. After aggregating the identified words from the sources and removing duplicates, we identified 215 unique keywords.

3.3.2 Keyword expansion. To identify potentially missing keywords, we loaded the GHTorrent export from March 2021 [37] in a local MySQL. We queried the `pull_request_comments` table for all the comments with at least one of our keywords. Among the 56.1 million pull request comments from GHTorrent, our search filtered a total of 26,307 comments with our keywords. In the next step, we wrote a Python script using the scikit-learn [49] library to compute the frequency of all the words in this corpus. After excluding our initial 215 keywords, common English stop-words, and words appearing in less than 100 comments, we created a list of 5,316 potentially missing words. Three authors independently went through this list to mark additional words for inclusion. In the next step, they had a joint discussion session to compare their individual lists and argue whether a word should be included. For the conflicting cases, they went with majority votes. At the end of the

Table 3. List of misogynistic keywords and their groups. Words in blue were added during our keyword expansion phase

Category (# of keywords)	Rationale	Keywords
Pejoratives (81)	Derogatory adjectives are often used to belittle women or express hostility [29, 43]	bitch, hoe, hysterical, uptight, slag, skank, slut, dimwit, whore, hormonal, chic, feminazi, chick, smug, cuck, horny, cocksucker, cougar, crone, skintern, bimbo, prostitute, harlot, heifer, gigolo, concubine, bawd, moll, floozy, cheater, witch, frump, wench, twat, sissy, mannish, flirty, ladylike, thot, cenobite, menstrual, vixen, kitten, hag, bossy, nagging, diva, mumsy, frumpy, cunt, feral, simp, fatcel, femcel, shrew, pickmeisha, foid, nympho, gold digger, promiscuity, puta, roastie, cock tease, milf, phony, mentalcel, psycho, conchuda, hustler, streetwalker, spinster, shrill, tart, karen, soccer mom, supermom, bastard, what she said, dowry bitchy, rep-whore
LGBTQ+ identities and slurs (30)	LGBTQ+ slurs or identities are often used to express hatred or demean this group	gay, lesbian, bisexual, transgender, queer, homosexual, gaydar, gaymer, gaysian, tommy, lesbo, sapphic, auntie, artiste, punk, batty, bufty, faggot, fag, pansy, dyke, tranny, trannie, ladyboy, dickgirl, sheman, shemale, transvestite, he-she, femboy
Uncomfortable reference (34)	Discussion of sexual acts in software development settings may put women or LGBTQ+ persons in an uncomfortable situation and are barriers to promoting inclusive discussions	boyfriend, penis, masturbate, dick, arse, asshole, carnal, erotic, genital, copulate, copulation, coitus, buttock, lovemaking, get laid, orgasm, virgin, make love, intercourse, menopause, porn, lust, libido, lewd, salacious, butt, smooch, kiss, naked, banging, balls, cum, cheating, lover
Women kins (15)	Mom jokes and citations of female relatives are found in content that expresses gendered insults	mother, mom, grandma, aunt, girlfriend, momma, mum, grandmother, granny, sister, niece, mommy, mummy, daughter, mama
Woman's body parts (15)	Women's body parts are often used for sexual objectification.	pussy, vagina, boob, tits, uterus, clitoris, clit, hymen, breast, nipple, ovary, areola, vulva, waist, lip
Women roles (14)	Traditional beliefs about gender roles also result in discrimination against women	wife, bride, actress, princess, waitress, queen, mistress, maid, nurse, housewife, heroin, nun, priestess, wives
General women (13)	Women specific words such as 'girl', 'women' or 'female' may be used to talk negatively or express 'stereotyping'	female, females, feminine, maternal, girl, herself, lady, gal, woman, women, pregnant, pregnancy, feminist, feminism, ladies
Flirtatious (25)	Words typically used in flirtatious contexts may allude to sexual harassment or unwanted sexual attentions	baby, tigress, doll, darling, honey, candy, dating, sweetie, cutie, sweetheart, babe, sugar, pretty, tootsie, romeo, marry, marriage, romantic, naughty, sexy, single, cute, relationship, propose
Physical appearance (13)	Words alluding to physical characteristics are often used to describe physical appearance	blonde, curvy, brunette, fugly, tomboy, busty, skinny, petite, butterface, blondie, fluffy, fat, hot
Sexual threat (9)	Words that express physical and sexual violence can be used to establish dominance and sexual harassment.	deflorate, hump, fornicate, molest, sodomize, rape, deflower, fornication, fuck
Cloth (11)	Clothes that are specific to women or LGBTQ+ persons can be used in for sexual objectification or demeaning	tampon, panty, panties, skirt, bikini, blouse, dress, lingerie, bra, leggings, sleeve
Men roles (5)	Reference to man relatives used for flirtation or dominance	dad, papa, daddy, father, husband

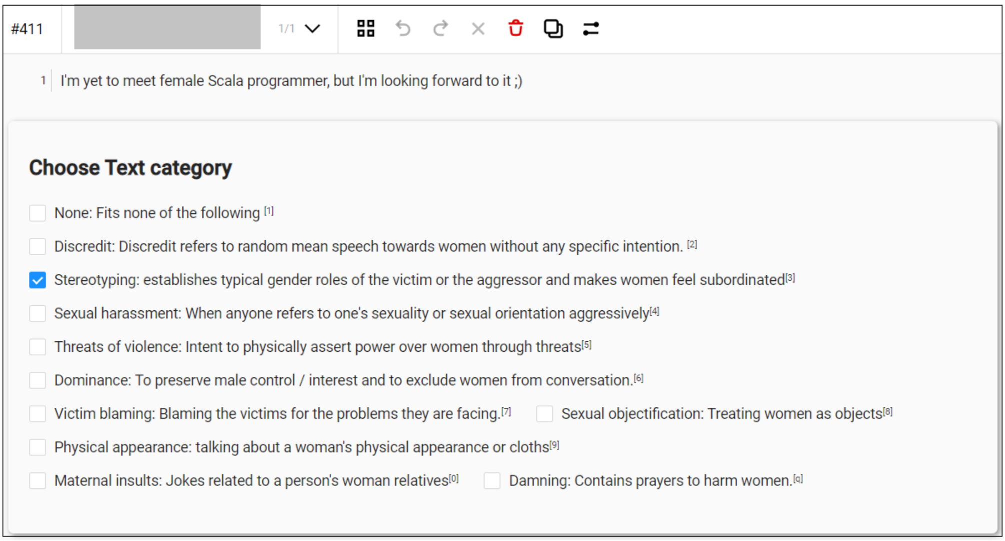
process, we identified 35 words to add to our keyword list. Keywords listed in blue in Table 3 are added during our keyword expansion phase. Although we did not include the 'Men roles' group among our initial 11 categories, we added this new group during this phase since words from this group may be used for flirtation or dominance (e.g., "Who is your daddy?").

3.3.3 Dataset Selection. We create a dataset of pull request (PR) comments on GitHub. We selected PR since it is a crucial mechanism to attract contributions from non-members and facilitate newcomers' onboarding among OSS projects [38]. PRs allow contributors to propose changes, which other community members then review. Due to the interpersonal nature of PR interactions and the potential for dissatisfaction due to unfavorable decisions, PR interactions may raise conflicts, frustrations, and incivility [31, 71]. We identified two possible options to curate a list of SGIDs from publicly available FLOSS projects on GitHub. Initially, we planned to use the GHTorrent MySQL dump [37] used for keyword expansion. However, we noticed three shortcomings with this approach. First, GHTorrent MySQL export truncates comments over 255 characters. Therefore, SGID excerpts beyond the 255-character boundary are unavailable in this dataset. Second, GHTorrent export does not include issue comments. Finally, we noticed a relatively smaller number of comments with LGBTQ+ slurs such as 'tranny' and 'faggot', and misogynistic pejoratives such as 'bimbo' and 'whore' in this dataset. However, our search using the GitHub search API returned a significantly higher number of such cases than those found in GHTorrent. This large discrepancy may also be because the most recent GHTorrent export was released over two years earlier. The second option is to mine directly from GitHub using its search API. We decided to use a hybrid approach using both the GitHub search API [4] and GHTorrent to mine the comments with each keyword. First, we mined all the comments with less than 255 characters and our 250 keywords from GHTorrent. We excluded comments of length 255, which may be truncated. Next, we queried the GitHub search API with our 250 keywords, limiting results to the first 1,000 entries. While it is possible to add additional filtering criteria, such as date range, to obtain more results beyond the first 1,000 entries from the GitHub search, we did not explore that study since our goal was to create a balanced dataset. As which categories of SGID a comment belongs to depends on the keywords contained, adding all comments with frequent keywords would create a more unbalanced dataset. After removing duplicates between the two datasets based on author, text, and timestamps, we had 225,117 unique comments after excluding bot-generated ones.

We also noticed that many comments are written in non-English, as many selected keywords have different meanings in other languages. Moreover, as we cannot comprehend texts primarily written in non-English, we decided to exclude those. We used fastText [17] to identify the probability each comment is written in English. After excluding all those with a probability of less than 50% being English, we had 193,056 comments.

However, this dataset is too large for manual labeling. To filter this dataset without discarding the ones more likely to be SGID, we applied a stratified sampling strategy proposed by Särndal et al. [73]. This strategy has also been used in prior studies [71, 72] where all the comments are classified using an off-the-shelf classifier that provides a probability of each comment being a positive instance. Then, all the comments with probability above a certain threshold are selected (i.e., Strata 1). To reduce tool bias, certain comments are randomly selected from those with probabilities below the threshold (i.e., Strata 2).

However, finding an off-the-shelf SGID tool was challenging. Despite many recent studies on Automated Misogyny Identification (AMI) [33], none of those for English is publicly available. Therefore, we selected a pre-trained BERT-based model proposed by Pamungkas *et* al. [62] to re-implement since this model boosted the second highest F1-score during their evaluation, and can be configured to output predictions as probability instead of binary classes opposed to their top



1 I'm yet to meet female Scala programmer, but I'm looking forward to it ;)

Choose Text category

□ None: Fits none of the following [1]

□ Discredit: Discredit refers to random mean speech towards women without any specific intention. [2]

☑ Stereotyping: establishes typical gender roles of the victim or the aggressor and makes women feel subordinated[3]

□ Sexual harassment: When anyone refers to one's sexuality or sexual orientation aggressively[4]

□ Threats of violence: Intent to physically assert power over women through threats[5]

□ Dominance: To preserve male control / interest and to exclude women from conversation.[6]

□ Victim blaming: Blaming the victims for the problems they are facing.[7] □ Sexual objectification: Treating women as objects[8]

□ Physical appearance: talking about a woman's physical appearance or cloths[9]

□ Maternal insults: Jokes related to a person's woman relatives[0] □ Damning: Contains prayers to harm women.[q]

Fig. 1. Interface of Label Studio for labeling one entry from the dataset

performing model. During our ten-fold cross-validation-based evaluations using the English AMI IberEval dataset [33], this model achieved 81.5% F1-score and 84.1% accuracy compared to 83.82% F1-score and 86.23% accuracy reported by the authors. We use this implementation (referred to as 'the *RefBERT*' hereinafter) to compute the probability that each comment of our dataset is an SGID.

We empirically determined the threshold for our sampling strategy. We wanted to put this threshold as low as possible to ensure we were not missing many SGID samples. We explored various options by lowering the threshold with 0.05 intervals starting at 0.5, the default threshold of RefBERT. With a threshold of 0.2, we obtained a dataset of 5,506 comments that have more than 20% probability of being an SGID according to the *RefBERT* [62]. We noticed that by lowering the threshold further to 0.15, we had to label an additional 3,683 instances manually. As manual labeling is highly time-consuming, we decided against further lowering. We term this dataset of 5,506 comments as our 'Dataset A: High probability with keywords', which comes from the Strata 1. We also selected samples from three other strata to mitigate tool biases (i.e., true positive cases potentially missed by RefBert), keyword biases (i.e., SGID cases with none of our keywords), and missing LGBTQ+ samples. To compensate for tool bias, we randomly selected an additional 2,501 comments (99% confidence interval and 2% margin of error [20]) from the remaining comments. We call this set 'Dataset B: Low probability with keywords.' To account for keyword biases (i.e., SGID cases with none of our keywords), we randomly selected 2,500 comments that do not include any of our 250 keywords. We termed this set as 'Dataset C: No keyword.' Finally, since we are using an AMI tool for stratified sampling, we noticed a lack of samples with LGBTQ+ words. Although our search found 6,254 comments with LGBTQ+ words, most are bot-generated spam. After filtering out the bot comments, we randomly selected 500 comments from the remaining ones. We call this set 'Dataset D: LGBTQ+ words.' Therefore, we selected a total of 11,007 unique comments for our dataset.

3.4 Step 4: Manual Labeling

To facilitate our manual labeling process, we installed an instance of *Label Studio*, the Open Source Data Labeling Tool [3] in a lab server. To assist with labeling, we configured the labeling interface to include a short definition for each category as listed in Table 1. The interface shows the raters one comment at a time and allows them to resume from the previous session. A rater could assign multiple categories to a comment.

Figure 1 shows an example from our labeling session. Three of the authors worked as raters in this stage. First, they sat together to discuss the inclusion/exclusion criteria and build an agreed-upon understanding. Then, they labeled the top 100 entries (i.e., according to RefBERT probability) from Dataset A through discussions to further validate their understanding. We created separate projects with the same dataset for different annotators to avoid biases in the labeling process. We divided the labeling tasks to ensure that at least two of the raters independently labeled each comment. We labeled the dataset in three iterations. First, the raters independently labeled 1,000 entries from Dataset A. We noticed still a high number of conflicts with a Krippendorff's α = *0.232* indicating a 'Low agreement'. They had another discussion session to resolve these conflicts and clarify misconceptions. While resolving conflicts after the first iteration, we found a few cases where comments were mislabeled due to a misunderstanding of the meaning of the texts or context. This session helped build a further improved understanding. We also noticed that binary labeling (i.e., whether a comment fits SGID or not) did not cause many conflicts; rather, which category or categories a comment fits was the primary reason. After that, they labeled the remaining 4,406 comments from Dataset A. This iteration achieved Krippendorff's α*=0.782* ('Acceptable agreement'). Again, we resolved the conflicts through mutual discussions. The third iteration had the rates independently label the remaining 5,501 instances (i.e., Datasets B, C, and D) and later held discussions to resolve conflicts. We achieved Krippendorff's α = *0.832* ('Satisfactory agreement') during this iteration. Overall, we achieved Cohen's κ = *0.658* for binary labeling (i.e., SGID and non-SGID) and Krippendorff's α = *0.691* for the 13-class labeling.

Due to the substantial size of our dataset, which consists of 11,007 instances extracted from pull request messages, conducting a thorough examination of the contexts for each entry would be a time-intensive process. Additionally, it is important to note that this study's primary goal is not an empirical investigation, and therefore, we focused on particular comments rather than delving into the specific contexts of those. It is worth recognizing that not all entries in the SGID class may exhibit discriminatory characteristics when the context is considered.

At the end of this labeling process, we identified a total of 1,422 (≈ 12.9%) SGID instances and 9,585 non-SGID instances. Table 4 shows the distribution of the final labels among the four datasets. Since some of the SGID comments belong to multiple categories, the sum of individual category counts is higher than the total number of SGID instances. The highest number of SGID entries from our dataset fit into *Sexual Objectification (SO)* category with 660 (≈6%) instances. *Anti-LGBTQ+* class ranks second with 339 (≈3%) cases. Next three classes based on number of instances are: *Maternal Insult* with 191 (≈1.73%), *Discredit* with 130 (≈1.2%), and *Appearance reference* with 81 (≈0.7%). We found only one instance for *Victim blaming*, making it the least common type of SGID in our dataset. However, we should be cautious to take this result as an accurate empirical distribution since our sampling method has limitations (i.e., 255 limit in GHTorrent and 1,000 results from GitHub search API). Guest *et* al.'s [40] dataset of 6,567 English Reddit posts includes 10.6% misogynous instances, whereas ours includes 12.9% SGID instances.

[3] https://labelstud.io/

Table 4. Distribution of different SGID and non-SGID classes among our labeled datasets

		Dataset				
Group	**Subcategory**	**A**	**B**	**C**	**D**	**Overall**
GSD	Discredit	120	9	0	1	130
	Stereotyping	24	0	0	0	24
	Sexual harassment	11	2	0	10	23
	Threats of violence	16	0	0	0	16
	Dominance	11	0	0	0	11
	Victim blaming	1	0	0	0	1
	Sexual objectification	605	46	0	9	660
	Appearance reference	76	4	0	1	81
	Maternal insults	172	9	0	10	191
	Damning	9	0	0	0	9
	Sexual reference	39	0	0	0	39
	Anti-LGBTQ+	16	1	0	322	339
	Others	72	2	0	0	74
	Total	1,019	65	0	338	1,422
Non-GSD		4,487	2,436	2,500	162	9,585
Total		5,506	2,501	2,500	500	11,007

3.5 Step 5: Evaluation of Existing Tools

First, we evaluated the performance of RefBERT [62] trained on the IberEval dataset to check the performance of an off-the-shelf model (i.e., trained on IberEval but evaluated on our dataset). Second, we retrained RefBERT [62] using our dataset to validate the need for a customized pipeline. Third, using our dataset, we retrained four closely related SE domain-specific NLP tools. Our selection of tools includes:

(1) ToxiCR [72] −is trained to identify toxic code reviews.
(2) STRUDEL tool [67] −is trained to identify toxicity from issue comments.
(3) Ferreira *et al.* [32] −is trained to identify incivility from issue comments.

Unsurprisingly, off-the-shelf RefBERT [62] performs poorly with only 0.46 precision, 0.342 recall, and 0.392 F1-score for the SGID class. These results validate the need to develop a domain-specific tool considering a domain-specific labeled dataset. After retraining with our dataset, RefBERT [62] shows significant performance improvements in precision (0.801), recall (0.693), and F1-score (0.743). Although retrained STRUDEL performs better than the off-the-shelf RefBERT [62], it fails to outperform RefBERT-retrained. This result may be due to the agnostic TFIDF vectorizer and the SVM algorithm used by STRUDEL, which tend to perform worse than transformers. Finally, retrained ToxiCR performs the best among the four tools with 0.815 precision, 0.703 recall, and 0.753 F1-score for the SGID class. These results suggest that SE domain-specific pre-processing helps improve performance for SE domain-specific NLP classifiers since RefBERT [62] and ToxiCR differ only in pre-processing steps. While all three models show promising results, their performances for the SGID class are significantly lower than those for the non-SGID class. However, such performance degradation for the minority class may not be surprising since our dataset is highly unbalanced, with approximately 1:6.7 ratio between the SGID and non-SGID samples. Since the identification of SGID samples is the primary goal of this study, we aim to build a custom tool that focuses

Table 5. Performance of the existing tools on our labeled dataset

Model	**Vectorizer**	**Non-SGID**			**SGID**			A	MCC
		P_0	R_0	$F1_0$	P_1	R_1	$F1_1$		
RefBERT (off-the-shelf)	BERT-base	0.901	0.941	0.923	0.460	0.342	0.392	0.863	0.321
RefBERT (retrain)	BERT-base	0.941	0.951	0.946	0.801	0.693	0.743	0.937	0.713
STRUDEL (retrain)	tfidf	0.958	0.955	0.956	0.701	0.715	0.707	0.924	0.664
Ferreira *et al.* [32] (retrain)	Bert-base	0.939	0.966	0.951	0.610	0.555	0.552	0.913	0.537
ToxiCR (retrain)	BERT-base	0.957	0.975	0.966	0.815	0.703	0.753	0.940	0.723

specifically on improving the performance of the SGID class. Since ToxiCR-retrain provides the best performance, we consider it the baseline for improving our proposed tool.

4 SGID4SE Design

The design of SGID4SE is motivated by prior SE domain-specific tools [7, 72] and builds on the ToxiCR framework [72]. The following subsections describe our dataset pre-processing, word vectorization techniques, an overview of the selected algorithms, and additional features added over the ToxiCR to improve performance for the SGID class.

4.1 Dataset pre-processing

Since pull requests and issue comments often contain URLs, word contractions, emoji markdowns, and source code snippets, we applied the following pre-processing steps to clean comments before using them for training. The first four steps are the same as ToxiCR [72]. The fifth step is customized to fit the SGID context, and the final step is specific to SGID4SE.

(1) *URL removal:* Many pull request comments contain URLs that refer to external posts or articles. We used a regular expression matcher to identify and remove the URLs.
(2) *Contraction expansion:* Contractions are the short form of one or multiple words. Developers use many contractions when communicating with each other. For example: shouldn't ⟶ should not, it's ⟶ it is. We replaced 153 common contractions with their expanded forms.
(3) *Special symbol removal:* We implemented a regular expression matcher to identify and remove special symbols (e.g., '&', '$').
(4) *Splitting identifiers:* During our labeling, we noticed examples of code snippets alluding to SGID categories (e.g., `queen.breastSize`). We used a regular expression matcher to identify and split identifiers written in *camelCase* or *under_score* forms. For example: `current_bride` ⟶ *current bride*, `breastSize` ⟶ *breast Size.*
(5) *Repetition elimination:* To avoid detection, we found instances of intentionally misspelled words. For example, we found one comment as *"Gaaaaaaaaaaaaaaaay Make it female version of johnson or something, janess?"* To identify such cases, we used a pattern-based matcher to identify and replace such cases with their respective intended forms.
(6) *Emoji removal:* Pull request comments contains emoji. Some of the emojis (i.e., ':ok_woman:') include words that can interfere with our classifier. We wrote a regular expression-based matcher to identify and replace emoji markups with a neutral word.

4.2 Algorithm Selection

Based on prior SE studies, we have evaluated three groups of algorithms in SGID4SE. Our selection includes four classical and ensemble (CLE) machine learning, three deep neural-network (DNN))-based, and three transformer-based algorithms. After excluding ToxiCR's two time-consuming yet low-performing algorithms BiLSTM and GBT, SGID4SE evaluates eight out of the ten ToxiCR selected algorithms [72]. Additionally, SGID4SE adds two pre-trained transformer-based models, ALBERT [53] and SBERT [68], since those have shown state-of-the-art performances for recent SE domain-specific sentiment analysis tasks [91].

- *Classic and Ensemble-based (CLE) models:* SGID4SE uses scikit-learn [65] implementations of the following four CLE algorithms: i) Decision Tree (DT), ii) Logistic Regression (LR), iii) Random Forest (RF), and iv) Support Vector Machine (SVM). We use the Term Frequency - Inverse Document Frequency (TF-IDF) vectorizer for the CLE models. Tf-IDf is a vectorization technique based on the Bag of Words (BOW) model that assesses the relevance of a word to a document within a collection and CLE models only works with Tf-IDf. The Tf-IDf score of a word is then calculated by multiplying its term frequency (Tf) by its inverse document frequency (Idf) as: $TfIdf(w,d) = Tf(w,d) * Idf(w)$. Here, $Tf(w,d)$ represents the term frequency (Tf), which is how often a word appears in a document, and the inverse document frequency (Idf), which measures the significance of the word across all documents. The formula for Tf is the frequency of the word in a document divided by the total number of words in that document. Idf is calculated using the formula: $Idf(w) = \log_e\left(\frac{N}{wN}\right)$. Here N is the total number of documents and wN is the number of documents containing the word.
- *Deep Neural Network (DNN) Models*: Using the TensorFlow library, SGID4SE implements three DNN models: i) Deep Pyramid CNN (DPCNN) [46], ii) Long Short Term Memory (LSTM) [39], and iii) Gated Recurrent Unit (GRU) [28]. We used pre-trained fastText embedding with these algorithms. FastText, developed by Facebook's AI team, is an efficient method for generating context-free word embeddings. FastText can handle out of vocabulary words by considering the morphological features of words. It creates a word's vector by combining vectors of its character substrings. As a result, it outperforms other word vectorization methods like Word2Vec and GloVe in natural language processing tasks, especially when the corpus contains unknown or rare words. Therefore, we have used FastText for DNN models. We chose the architecture of the selected models from existing text classifiers [28, 46, 52, 72].
- *Pre-trained Transformer Models (PTM)*: Using tensorflow_hub, we used pre-trained BERT encoders. We have used three different BERT models. i) bert_en_uncased, commonly referred as BERT-base model, which is trained with 2,500 million words from Wikipedia and 800 million words from the book corpus; ii) a Lite BERT aka ALBERT base [53]; and iii) a BERT experts model (SBERT), customized using the Stanford Sentiment Treebank (SST-2) for sentiment analysis tasks.

4.3 Performance Improvement for the Minority Class

ToxiCR does not implement any strategy to improve the performance of the minority class to encounter unbalanced training data. As our SGID dataset is more unbalanced than ToxiCR's training dataset, SGID4SE implements the following three mitigation strategies. We did not use undersampling since it can result in loss of information if the training dataset is not very large [41].

- *Higher weight for the minority class:* With this strategy, no samples are duplicated or excluded. During the training, features belonging to the samples from the minority class get higher weights (usually a fixed multiplier sent as a parameter) than those from the majority [18]. While an oversampling strategy increases training time, this strategy does not cause such overhead. However, this strategy cannot be applied to all algorithms. While all the neural network and transformer-based algorithms support this customization, only *RandomForest* from the CLE group supports it.

- *Random oversampling:* In this strategy, randomly selected instances from the minority class are duplicated until a desired ratio between the two classes (i.e., SGID: non-SGID) is achieved [41, 92].
- *Word replacement based new samples:* In this strategy, we manually grouped the keywords from Table 3 into 44 GSID equivalent groups. We consider two words belonging to the same group if replacing one with the other in an SGID comment results in a new SGID comment. While the common strategy is to consider only synonyms to generate new samples based on word replacement, we noticed that two words can be equivalent in SGID contexts even if they are not synonyms. For example, while 'mom' and 'grandma' are not synonyms, replacing the former in 'This is cheating harder than your mom does.' – creates another SGID. Therefore, the following nine words – "mother", "mom", "momma", "mommy", "mummy", "mama", "grandma", "granny", and "grandmother", belong to the same SGID group. Therefore, the word replacement strategy would create eight new samples if the original one includes the word "mom". To create equivalent groups, we started with 12 categories from Table 3. If two words belonging to the same cannot be equivalently replaced by one another in an SGID comment from our dataset, we placed them in two different equivalent groups. In the end, we created 44 GSID equivalent groups from the initial 12. Our replication package includes the list of words belonging to each group.
  Using this strategy, SGID4SE automatically generates additional SGID samples from the ones from the training set. During training, randomly selected SGID instances are added to the training set based on the desired ratio between the two classes. For example, if the training set includes x SGID, y non-SGID, and the desired ratio is 0.5, then $(0.5 * y - x)$ generated samples are randomly selected and added to the training set.
- *Mixed sampling:* This sampling strategy is a combination of Random and Word replacement-based ones [41], where half of the additional samples to reach the desired ratio are duplicates, and the remaining half are generated ones.

4.4 Optimal Threshold Identification

During predictions, DNN-based classifiers output the probability of a sample belonging to a particular class. These output probabilities are converted into binary by comparing them against a pre-defined threshold. Many classifiers, including ToxiCR, use 0.5 as the threshold for such conversion. However, recent results suggest that 0.5 may not be the optimum choice, and a model can achieve improved performance by empirically evaluating this parameter [13, 70]. SGID4SE implements a feature to automatically identify the threshold value that provides the best performance on the test dataset during 10-fold cross-validations of DNN and PTM models.

4.5 Word count based features

We implemented word count-based features suggested by Pamungkas *et* al. [62] in SGID4SE. We modified the pre-processing pipeline to count the number of keywords belonging to seven out of the 12 categories from Table 3. We excluded five categories, as we found that keywords from those do not appear frequently among SGID samples. The selected seven categories are: 1) pejorative, 2) women relatives, 3) LGBTQ+ identities and slurs, 4) women body parts, 5) women roles, 6) women-specific clothes, and 7) women roles. If enabled, SGID4SE adds an additional seven dimensions to include these word counts after a comment is vectorized.

Table 6. Performance of the algorithms without any performance optimization in SGID4SE. A shaded background indicates significant improvements over the baseline model's performance, i.e., ToxiCR(retrain) for a metric.

Group	Algo	Vectorizer	Non-SGID			SGID			A	MCC
			P_0	R_0	$F1_0$	P_1	R_1	$F1_1$		
CLE	DT	tfidf	0.941	0.956	0.948	0.670	0.598	0.631	0.910	0.581
	LR	tfidf	0.924	**0.997**	0.959	**0.957**	0.449	0.610	0.926	0.627
	RF	tfidf	0.940	0.984	0.961	0.845	0.575	0.683	0.931	0.662
	SVM	tfidf	0.935	0.992	0.963	0.911	0.536	0.674	0.933	0.668
DNN	CNN	fasttext	0.954	0.983	**0.968**	0.870	0.680	0.756	**0.944**	**0.736**
	GRU	fasttext	0.957	0.978	0.967	0.829	0.703	**0.761**	0.943	0.732
	LSTM	fasttext	0.948	0.985	0.966	0.867	0.634	0.730	0.940	0.709
PTM	ALBERT	bert	0.945	0.966	0.955	0.743	0.618	0.667	0.921	0.631
	BERT	bert-base	**0.960**	0.973	0.966	0.806	**0.722**	0.758	0.941	0.728
	SBERT	bert	0.959	0.969	0.964	0.776	0.721	0.745	0.937	0.711

5 Training and Evaluation

The following subsections detail our training and evaluation of SGID4SE.

5.1 Evaluation Configuration

We have used five metrics for evaluation: i) Precision: the percentage of identified cases that belong to that class, ii) Recall: the ratio of the correctly predicted and total number of cases, iii) F1-score: the harmonic mean of precision and recall, iv) Accuracy: the percentage of cases that a model predicted correctly, and v) MCC: it considers the true positives, false positives, false negatives, and true negatives from the confusion matrix and calculates the correlation between the predicted class and true class. The MCC score ranges from -1 to 1 and is considered a more balanced measure than accuracy and F1-score for evaluating binary classification tasks [21]. In our evaluations, we consider MCC as the most important metric to evaluate these models since it is considered the most balanced measure. In case of an MCC tie, we consider the F1-score for the SGID class since: i) Identification of SGID comments is our primary objective, and ii) our dataset is imbalanced with 85% non-SGID comments. We evaluated the CLE models using 10-fold cross-validation, where the dataset was split into ten random subsets. Each of the sets was used exactly once for testing, while the remaining nine sets were used for training the model. We computed the average for all eight (i.e., precision, recall, and F1-scores are computed separately for both classes) metrics over the ten runs. For the DNN and Transformer models, our dataset is split into an 8:1:1 ratio where eight sets are used for training, 1 for validation during training, and the remaining for testing. The validation set helps to optimize the hyper-parameters and prevent the model from over-fitting. We set the number of epochs as 30 during training. The validation set is monitored using `EarlyStopping` function for the validation loss measure. We set the `patience` parameter to 4 and `restore_best_weights=True`. This stops models early after a minimum validation loss is achieved, and four consecutive runs do not achieve further lower loss. Weights for the model with the minimum loss are used for testing. As the model performances are normally distributed, we use paired sample t-tests to check if observed performance differences between two configurations are statistically significant ($p < 0.05$). We use the 'paired sample t-test' since we initialize the random number generator using the same seed to guarantee that cross-validation runs would get the same train/test partition sequences.

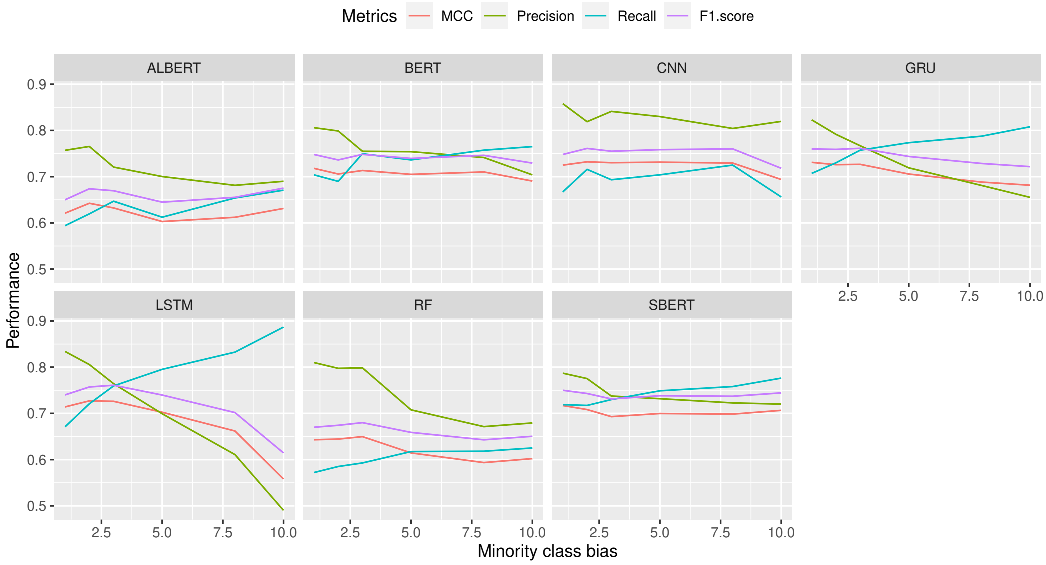


Fig. 2. The impacts of minority class weight variations on precision, recall, and F1-score for the SGID class and MCC

5.2 How does each algorithm perform in its basic configuration?

Table 6 shows the performances of the ten selected models with our dataset without any optimization steps. We have highlighted cells with scores that significantly outperform (i.e., $p < 0.05$ according to the result of a t-test) our best baseline model (i.e., retrained ToxiCR shown in Table 5). The results suggest that CNN, GRU, and BERT significantly outperform our baseline model in terms of MCC and $F1$ score. BERT achieves the best recall among the ten algorithms while also outperforming ToxiCR-retrain's BERT for five out of the eight metrics. Since SGID4SE's basic-configuration BERT differs from ToxiCR-retrain's implementation in terms of two pre-processing steps as described in Section 4.1, we can attribute these improvements to those two steps.

Key finding 1: *Three of the ten algorithms significantly outperform the established baseline regarding the two key metrics (i.e., the F1-score for the SGID class and MCC). Two additional data pre-processing steps (repetition elimination and emoji removal) implemented in SGID4SE provide significant performance boosts for key metrics.*

5.3 Does assigning higher weights to minority class samples during training provide a performance boost?

Figure 2 shows the variations of precision, recall, and F1-score for the SGID class and MCC with increasing minority sample weights. As LR, DT, and SVM lack support for class weighting parameters, this analysis was conducted on the remaining seven algorithms. We considered six different class weights for the SGID classes: 2, 3, 5, 8, and 10, with 1 serving as the baseline. As expected, precision drops when minority samples get additional weights. For ALBERT, BERT, GRU, LSTM, RF, and SBERT, the recall for the SGID class increases with higher class weights, except for CNN, which experiences a decrease in recall from 66.7% to 65.6% with greater class weights. Regarding F1 score, as the class weight increases, the performance of BERT, CNN, LSTM, and

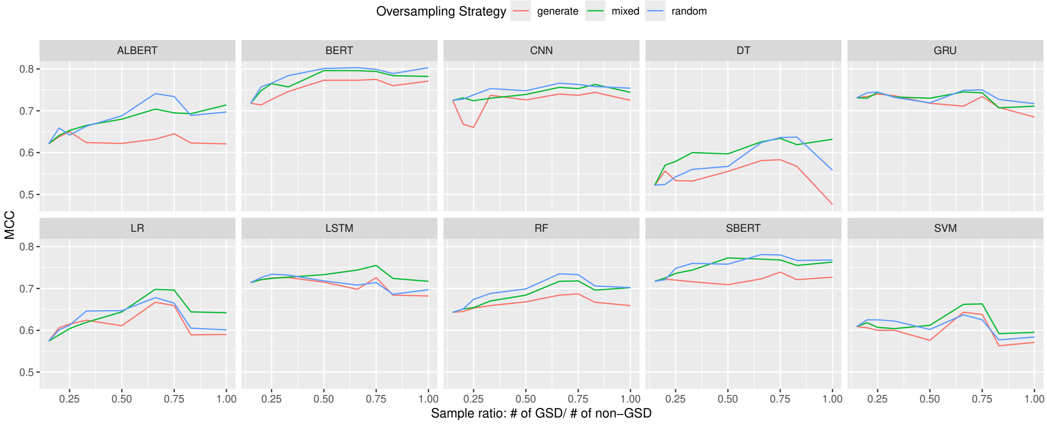


Fig. 3. MCC change with variation of sampling strategy/ratio

SBERT deteriorates, whereas ALBERT exhibits a 2% improvement, GRU shows a 10% increase, and RF demonstrates a 5% increase. However, contrary to our expectations, recall gains are slower compared to the drops in precision. Hence, with this strategy, we see either drops or stagnant MCC and $F1$-score. Only ALBERT exhibits enhanced MCC with a class weight of 10 for the SGID class, while all other models perform significantly (paired sample t-test, $p < 0.05$) worse than the baseline class weight. Therefore, this strategy is not viable for achieving higher performance on our dataset.

Key finding 2: *We could not achieve significant (paired sample t-test) performance boosts for the key metrics (i.e., MCC) by assigning higher weights to the SGID samples during training on our dataset.*

5.4 Which one is the best oversampling ratio/strategy for each algorithm?

Our models exhibit a bias towards the majority class due to imbalanced SGID and non-SGID ratios, potentially resulting in the neglect of minority class features because of their under-representation. We assessed three oversampling strategies to mitigate such biases: i) random oversampling, ii) generation-based oversampling, and iii) a mixed approach. Figure 3 illustrates how the MCC of the models vary based on different oversampling methods and ratios, including 0.15, 0.20, 0.25, 0.33, 0.50, 0.66, 0.75, 0.83, and 1. The 0.15:1 ratio serves as the baseline, as that is the original ratio between the two classes in our dataset. On the one hand, we noticed that the random approach provided the best MCC for a particular ratio for seven out of ten algorithms. In contrast, the 'mixed' approach ranked second, and the 'generate' approach performed the worst. More importantly, we found that, for each algorithm, the model with optimum oversampling configuration had significantly higher MCC than its baseline performance reported in Table 5.2. Table 8 shows the best oversampling approach/ratio configuration for the ten algorithms on our dataset.

Key finding 3: *Both random and mixed approaches are viable options to achieve higher MCC scores on our dataset, where random provides the best performance for seven out of the ten models. In contrast, the optimum ratio varies between 0.66 to 1 among the algorithms, with 0.66 being the optimum one for five.*

Table 7. Performances of the models with word count-based features. A shaded background indicates significant improvements achieved through word-count features.

Group	Algo	Non-SGID			SGID			A	MCC
		P_0	R_0	$F1_0$	P_1	R_1	$F1_1$		
CLE	DT	0.946	0.955	0.951	0.677	0.634	0.654	0.913	0.606
	LR	0.933	0.992	0.962	0.909	0.522	0.663	0.931	0.658
	RF	0.946	0.985	0.965	0.863	0.623	0.723	0.938	0.701
	SVM	0.942	0.987	0.964	0.872	0.588	0.702	0.936	0.684
DNN	CNN	0.957	0.978	0.967	0.837	0.706	0.762	0.943	0.735
	GRU	0.956	0.978	0.967	0.828	0.693	0.753	0.942	0.725
	LSTM	0.950	0.984	0.967	0.865	0.654	0.743	0.941	0.720
PTM	ALBERT	0.947	0.973	0.959	0.785	0.627	0.688	0.928	0.659
	BERT	0.958	0.970	0.964	0.784	0.711	0.743	0.937	0.710
	SBERT	0.959	0.969	0.964	0.777	0.717	0.744	0.936	0.710

5.5 Does optional word count-based features help achieve higher performances?

Table 7 shows the performance of the models when word-count-based features are enabled. We also performed pair-sampled t-tests to check if performance improvements (if any) are statistically significant and mark such cases with shaded backgrounds in Table 7. As per the result, word count-based features significantly improved the performances of all four CLE algorithms.

> **Key finding 4:** *Word count-based features significantly improve the performances of all CLE models but do not improve performances for DNN or PTM models on our SGID dataset.*

5.6 How does each algorithm perform with an optimum configuration?

Table 8. Performance of the models with the best combination of word-count features, oversampling method, and the ratio between SGID and non-SGID

Algo	Best Configuration			Non-SGID			SGID			A	MCC
	Word count	Oversample approach	#non-SGID / # SGID	P_0	R_0	$F1_0$	P_1	R_1	$F1_1$		
DT	✓	random	0.83	0.957	0.944	0.951	0.656	0.717	0.685	0.915	0.637
LR	✓	mixed	0.66	0.954	0.973	0.963	0.792	0.681	0.732	0.935	0.698
RF	✓	random	0.66	0.957	0.980	0.968	0.836	0.703	0.763	0.944	0.735
SVM	✓	mixed	0.75	0.957	0.955	0.956	0.700	0.714	0.707	0.924	0.663
CNN	*X*	random	0.66	0.957	**0.987**	0.972	**0.889**	0.702	0.783	0.950	0.763
GRU	*X*	random	0.75	0.959	0.980	0.970	0.851	0.714	0.776	0.947	0.750
LSTM	*X*	mixed	0.75	0.969	0.966	0.968	0.782	0.794	0.786	0.944	0.755
ALBERT	*X*	random	0.66	0.962	0.973	0.968	0.803	0.743	0.772	0.943	0.738
BERT	*X*	random	1	0.971	0.980	**0.975**	0.859	0.800	**0.828**	**0.957**	0.804
SBERT	*X*	random	0.66	**0.972**	0.971	0.971	0.810	**0.811**	0.808	0.950	0.781

Table 8 shows the performances of the ten models with optimum configurations that we found based on our evaluation of sampling strategy/ratio and word count feature evaluations. We have also highlighted the best value for each metric with bold letters. Furthermore, we have shaded the cells in gray to indicate results that are significantly better than the baseline model ToxiCR (based on t-tests with p-values < 0.05) We found BERT emerging as the top-performing model, with five of its measures having the best scores, which include precision for non-SGID class, F1-scores

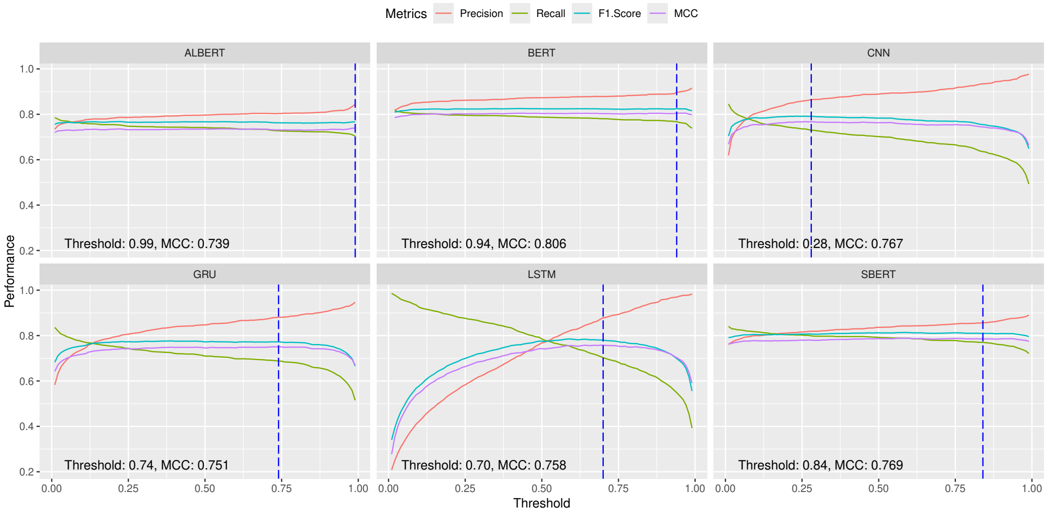


Fig. 4. Performance improvement with threshold variation.

Table 9. Performances of the DNN and PTM models with the threshold to maximize MCC. A shaded background indicates significant improvements over the model with the default threshold (0.5).

Algo.	Threshold	Non-SGID			SGID			A	MCC
		P_0	R_0	$F1_0$	P_1	R_1	$F1_1$		
CNN	0.28	0.961	0.983	0.972	0.865	0.730	0.791	0.950	0.767
GRU	0.74	0.955	0.986	0.970	0.881	0.688	0.772	0.948	0.751
LSTM	0.70	0.957	0.985	0.971	0.878	0.703	0.780	0.949	0.758
ALBERT	0.99	0.957	0.980	0.969	0.843	0.704	0.766	0.945	0.739
BERT	0.94	0.968	0.984	0.976	0.883	0.782	0.828	0.958	0.807
SBERT	0.84	0.966	0.981	0.973	0.857	0.770	0.810	0.953	0.786

for both classes, MCC, and accuracy. SBERT remains the second-best model in terms of MCC and F1-score. All three DNN models and two of the three PTM models, excluding ALBERT, outperform ToxiCR-retrain in terms of the two key metrics, MCC and F1-score for the SGID class.

> **Key finding 5:** *BERT boosts the best performance with 85.9% precision, 80.0% recall, 82.8% F1-score, 95.7% accuracy, and 80.4% MCC with random-oversampling. Five neural network-based models outperform the baseline ToxiCR-retrain.*

5.7 Do optimal threshold selection improve performance for the best models?

Since only DNN and PTM support this parameter, this analysis only applies to the six algorithms belonging to those groups. With the optimum configurations from Table 8, we varied the threshold from 0.1 to 0.99 and computed ten-fold cross-validation performances for each threshold. Figure 4 shows variation in performances for the SGID class and MCC score with varying threshold and optimum threshold values. We noticed that the optimum threshold for all six models is higher than the default value (0.5). We also noticed that DNN models are more sensitive to thresholds than the PTMs and encounter wider variations. Table 9 shows the performances of the DNN and PTM

Table 10. Confusion Matrix for our best-performing model (i.e., BERT)

		Predicted	
		SGID	**Non-SGID**
Actual	**SGID**	1, 138	286
	Non-GSD	190	9, 393

models with empirically determined optimum thresholds. We also performed 'pair sampled t-tests' to identify statistically significant improvements that we show with shaded backgrounds. Notably, we noticed significant improvements in precision for the SGID class for five of the six models. However, we did not notice significant improvements in the two key metrics, MCC and F1-score of the SGID class. While BERT with a 0.98 threshold provides the best MCC, its improvement over the model with a default threshold is not statistically significant. Therefore, we still consider BERT with a 0.5 threshold as the best-performing model and report this model's performance as the best one throughout the paper.

Key finding 6: *While empirically identified optimum threshold improves MCC, F1-score of the SGID class, and MCC, these improvements are not statistically significant. However, optimum thresholds do significantly improve precisions of the SGID class.*

5.8 What are the most common misclassifications from the best-performing model?

To better understand our model, we have analyzed the misclassifications of the best-performing (i.e., BERT with a 0.5 threshold) model. Table 10 shows the confusion matrix of this model, which misclassifies 476 comments out of 11,007 comments, with 190 false positives (FP) and 286 false negatives (FN). We followed an open coding approach to categorize the misclassified ones. After independently scrutinizing misclassified ones, two authors discussed these errors to create a classification scheme for false positives with four categories. Then, these raters independently tagged each misclassified case with one of those four categories. They compared the assigned tags and resolved conflicts through another discussion. For the False negatives, we looked into our 12 class SGID schema from Table 1.

False positives (FP).

(1) *Project Discussion (PD):* GitHub hosts many 'R-rated' open source games. Pull request discussions for such projects include sexually explicit language as game scenarios. For example, *"suggestion: If the butcher didn't cut off the skin, the tits here are more recognizable."*, is suggesting to change the message shown. We did not label such cases as SGIDs, but our classifier predicted those as ones. Overall, 33% of the FPs belong to this category.

(2) *Pre-processing induced errors:* We implemented regular expressions to identify adversarial patterns where one or two characters are intentionally misplaced or repeated (e.g., 'b00b', and 'gaaay') to avoid getting flagged. While these patterns were successful in fixing most of such cases, those also replaced some of the benign phrases (e.g., 'booboo', and 'boob') with SGID ones. For instance, *"Can i keep the honey boo boo?"*. Around 11.8% of FPs belonged to this category.

(3) *General Error (GE):* The number of SGID instances in our dataset is relatively small. Therefore, our dataset does not have adequate samples to build a highly precise classifier. As a result, our model misclassifies some comments as SGID if those contain words from the 'General women' group (see Table 3). We label such errors as GE, which includes 51.13% of the FP cases. For example, *"One of my random blonde moments; will fix."* is incorrectly classified as an SGID by our model.

(4) *Toxic but not SGID:* Although toxic, some comments do not fit the criteria to get labeled as SGID. For example, *"Please make their yield be five so they don't fucked over by Rng"* is toxic but not SGID. Almost 4.07% FPs belong to this group.

False Negatives (FN). We also analyzed the FN cases to identify which SGID categories were more frequently missed by our model. Not only is our dataset imbalanced, but approximately 50% of the SGID instances belong to the sexual objectification category (See Table 4). We noticed that categories with fewer instances were more likely to be missed. For example, we had only 23 sexual harassment cases, and our model failed to identify 8 out of those 23 with a false negative rate of 34.7%. We found 56.15% false negatives for 'Discredit' despite this category having 130 instances and 41.67% of FNs for 'Stereotyping'. Pre-processing steps targeting this category can help train a better SGID classifier using our dataset. The lowest false negative rate (i.e., 8%) is seen for the anti-LGBTQ+ class, as most of the instances from this class contain anti-LGBTQ+ slurs. Both 'Maternal insults' (3.14%) and 'Sexual objectification' (11%) had lower false negatives than the overall model, as each of those classes had more than 100 instances, and identifying those cases can be straightforward because of the presence of SGID-related keywords.

Key finding 7: *Presence of keywords associated with gender groups such as 'woman', 'girl', 'female', 'females' are more commonly associated with false positives. The best model frequently misses instances belonging to Discredit and Stereotyping.*

6 Discussion

The following sections discuss key lessons based on our study and provide several directions for future research.

6.1 Findings:

The following are the key findings based on this study.

Lesson #1: SGIDs on GitHub differ from social media.: Sexual harassment is one of the most frequent categories of SGID on social media, with more than one-fifth of the cases [14, 33]. However, we noticed less than 1.7% sexual harassment (23) among our SGID cases. Among the 1,422 SGID instances, SO (≈46%) and Anti-LGBTQ+ (≈23.8%) are the most frequent ones. Prior studies show a dominance of particular categories of misogyny in different languages. A strong representation of *'neosexism'* was found in Danish tweets where discrimination against women is questioned and men are presented as victims [90]. Mulki *et* al. [58] introduced a new category named *'damning'* while labeling misogynistic content in Arabic tweets. Though it is unclear whether language, context, or culture impact the dominance of a particular type of SGID, it requires further investigation.

Similar to existing lexicon-based data collection studies [29], our study may suffer from a lack of completeness because of a limited collection of keywords. From the labeled SGID texts, we can target a few projects where SGID-positive comments were generated and explore the existence of SGIDs in depth. That will assist in mitigating lexicon-based bias and increase the variety of linguistics [40]. However, almost 73% instances belonging to three groups imply sexual objectification, anti-LGBTQ+ comments, and discredits are the dominant forms of SGIDs on GitHub. These findings also align with prior FLOSS studies [77, 78].

On the other hand, some of the interactions that may seem non-SGID during general conversations are more likely to be SGIDs during pull request contexts. For example, we noticed maternal insults (aka 'mom jokes') as the third most frequent SGID category. These cases often represent women as computer illiterate, nagging, or sexual objects. For example, we found a pull request comment saying, "okay, mom!". Upon further investigation, we found that the author complained about a reviewer's insistence on some changes. However, in a general context, this text is not an SGID. But in this context, it implicitly says "okay, [you are nagging like a] mom!". Therefore, in this context, it can be considered an SGID.

Lesson#2: Existing toxicity detector tools do not perform well for SGID content. We evaluated the performance of STRUDEL and ToxiCR tools on our dataset and found that they did not yield satisfactory results in detecting SGID content where ToxiCR gives the best precision as 81.5%. However, GRU and BERT exhibited superior performance compared to these tools, even without any performance-enhancing strategies. Notably, after fine-tuning with an oversampling method, BERT achieved an impressive 95.7% accuracy and a precision of 84.1%. Consequently, our findings underscore the need for a dedicated tool tailored to the task of SGID content identification.

Lesson #3: Application of SGID4SE to combat SGIDs. BERT outperformed all other models with the best performance with 85.9% precision, 80.0% recall, 82.8% F1-score, 95.7% accuracy, and 80.4% MCC with random-oversampling technique. Future researchers should consider this model as a baseline for their studies. However, since both the precision and recall of our best model are close to 80%, it misses approximately one in five SGID cases. While we concur that significant improvement opportunities remain for SE domain-specific SGID models, a project can utilize our model to automatically flag potential SGIDs. A project administrator can review a flagged communication and inspect the context to make a final determination. SGID4SE's 86% precision indicates that one out of the seven flagged cases is more likely to be a false negative.

Lesson #4: Games-related projects harbor SGIDs due to the target audience. The #gamergate brought attention to sexism and misogyny in the gaming community [55]. We also noticed many comments among gaming projects that include SGID words. However, in most of those cases, we found those as character dialogues. Considering the character quotes, we labeled those as non-SGIDs. However, many such cases appeared as false positives by our classifiers.

Lesson #5: Regulations are necessary for naming conventions. We found the usernames of contributors that contain misogynistic keywords. For instance, *hornygranny*. It should be enforced that users must avoid misogynistic words while creating usernames, variable names, or library names. Another instance, *"Actually, this is still wrong, since 'listOfToxinsInThisBitch' is a list of types, the for loop will filter everything out."* Using such sexist words in variable names may create discomfort for women developers. Also, many libraries are named with misogynistic keywords. For instance, *"s/thosecunts/helpdesk/g"*. Not surprisingly, due to the absence of women, misogynistic keywords are used in naming libraries or resources. Therefore, CoC should be enforced in naming conventions.

Lesson #6: Educating developers regarding SGIDs and their implications. A few comments in our dataset report about stereotyping or sexist behavior and the limited capability of developers to take action against those. For example, a pull request comment says, *"I've found that people are ignorant, ill-informed, or against diversity. So often, I encounter 'there is nothing we can do,' 'it is not worth the effort,' 'women just don't want to work with computers,' or 'women will end up having babies so ...'"*. This comment shows a developer being concerned about SGIDs and initiating a discussion in a pull request comment. Education materials can be created to help developers better understand why a particular interaction is SGID and should not be used.

6.2 Directions for future researchers:

While crafting the tool, we have gained following insights that could benefit future designers working on SGID content detection tools.

Insight #1: Investigation of context and sexist roots of words and phrases is necessary for accurate empirical investigation. Understanding the sexist roots of different terms and phrases might be challenging. For example, *"ok soccer mom gosh"*. Here, calling a person 'soccer mom' means that person is being called insistent and super busy. It is a stereotyping that may be acceptable in other discussions but not in the SE context. For another instance, *"should we really be eating power puff girl chili?"*. Here, "power puff girl" refers to a cartoon where superhero characters are girls, and 'chili' means pepper. So the person who is asking will be doing something "difficult" based on someone's advice (maybe a woman). Moreover, it is difficult to understand the intended meaning accurately without knowing the author and the target. For example, *"you are aboslut right ;)"*. The wink emoticon at the end of the text might express that the typo is intentional and the author is trying to convey something with a different meaning. 'Karen' is another stereotyping word used for white women. We noticed a couple of instances during our labeling. Although we labeled those as 'SGID's, the target person's name may be 'Karen.'

Insight #2: Domain-specific customization is necessary. Misogyny identification is a natural language processing (NLP) task with a limited number of labeled datasets. To the best of our knowledge, ours is the first one from the SE domain. However, during our first labeling iteration, the annotators had difficulty making decisions since many phrases may convey different meanings for SE context, for example, *"That's good practice. _Skinny views, fat models_ they say."* For a non-SE person, this text is talking about women, but in the SE domain, it discusses project components using the Model-View-Controller (MVC) architecture. Therefore, an SE domain-specific tool is necessary.

Insight #3: Addressing typos/ misspelled misogynistic keywords is necessary. While filtering the contents based on keywords, we found that many texts contain sexist keywords that may have happened due to typos. For example, hoe → how, flag → fag, busy → bussy, and witch → which. While it is difficult to discern whether those typos were intentional, we put those instances into the non-SGID group. Future researchers should keep an eye on such cases while building SGID tools. On the other hand, someone can alter or replace letters, for example, o with 0 or l with 1, intentionally to circumvent the automatic detection of sexist content. We used regular expression matching during the pre-processing step to identify such cases. However, these pre-processing steps also introduced several false positives. Automated identification of such adversarial examples remains a challenge.

Insight #4: More SGIDs, especially the ones from the less represented groups, are needed to improve performance. Despite our labor-intensive keyword list preparation, stratified sampling strategy, and labeling more than 11K instances, we found only 1,422 SGIDs. Despite these challenges, the precision and recall of our best model are close to the state-of-the-art ones achieved on Twitter or Reddit datasets [62]. Our error analysis suggests that lower representations from five out of the twelve SGID groups also contribute to false negatives. We noticed that 32% of our keywords fall under the umbrella of 'Pejoratives' (Table 3). Texts, including pejorative, are more likely to fall into either SO, Stereotyping, or Discredit. It might be a reason for the prominence of those three classes in our dataset. Therefore, adding more lexicons to other groups might help to find cases belonging to those. While a multiclass classifier to identify which categories of SGID a text falls under is the ultimate goal for this research, more work needs to be done to accomplish that goal. Since we have 12 classes, multiclass classification is extremely challenging. Moreover, our dataset is highly unbalanced, with most instances belonging to four categories. Hence, rarely represented classes would have poor performance with this dataset. Therefore, we would require a new strategy to identify adequate instances for underrepresented classes.

Insight #5: Customized preprocessing steps is crucial. It is crucial to recognize that gender discrimination and other forms of toxicity and incivility come in various shapes and sizes. Therefore, customizing pre-processing steps is essential. For instance, you should employ a pattern-based matcher to pinpoint SGID-related keywords while also implementing domain-specific pre-processing steps for developers, such as splitting identifiers. Additionally, it's worth noting that incorporating optional counting-based features can potentially enhance the performance of CLE models. However, regarding DNNs and PTMs, these features might not yield the same performance boost. We also discovered numerous texts containing sexist keywords related to a game or project scenario that we did not classify as SGID content. Future tool developers should remain vigilant for this type of material. Additionally, we encountered many false negatives in the 'Discredit' and 'Stereotyping' categories. More detailed preprocessing steps focused on these categories can be introduced to enhance performance.

7 Threats to Validity

Internal validity: Our keyword-based filtering is the primary source of internal validity. We curated our list of keywords from multiple sources. We also included a keyword expansion step to identify potentially missing keywords. Yet, the lack of completeness of keywords remains a concern. We tried to mitigate this threat by randomly selecting 2,500 instances (i.e., Dataset C) that do not include our keywords. Since we did not find any SGID in Dataset C, the likelihood of missing a very large number of SGIDs due to the incompleteness of our keywords is minimal. Our stratified sampling using an off-the-shelf AMI tool may also introduce a bias if it performs better for certain classes of SGIDs. We mitigated this threat using a very low threshold (i.e., 0.2) and also including a sample of 2501 texts below this threshold.

Construct validity: Potential annotator biases are primary construct validity threats for this study. Our team of annotators includes two women and two men, all aged between 20-35. However, we did not rely on the gender diversity of our team and took precautions to avoid personal biases. We followed rubrics from peer-reviewed studies [40, 78, 81]. We annotated our rubric with examples and the annotators spent a significant amount of time discussing the rubric and talking about hypothetical positive and negative cases to build a common understanding. Moreover, we labeled the rubrics in three phases to improve their understanding based on new types of cases that they may have not seen in prior sets. Although a few instances from our study may still be subject to annotator biases, the number of such cases may not be large enough to invalidate our results, due to our carefully designed rubric and labeling protocol. Moreover, we mostly retained the default hyperparameters for the CLE algorithms and did not make significant adjustments. For ToxiCR [72], researchers explored six parameters with a total of 5,040 combinations for RandomForest and five parameters with 360 combinations for DecisionTree and concluded that hyperparameter tuning did not lead to notable performance improvements for these algorithms. Also, due to significant computational costs, they did not conduct the hyperparameter tuning for the DNN algorithms too. Based on their findings, we opted not to perform extensive hyperparameter tuning for our SGID4SE tool.

Conclusion validity: Since the primary objective of this study is to develop an automated model with a diverse set of instances, we decided to search from all GitHub communications. However, due to technical limitations such as 255 characters limit in GHTorrent and 1,000 results per query from GitHub search API, we may have missed a large number of SGID comments, especially the ones with words appearing frequently appearing in non-SGID contexts (e.g., 'girl', 'mother', and 'sexy'). Therefore, the distributions of SGIDs shown in Table 4 may not be an accurate representation. To identify more accurate distributions, we would need to carefully curate a sample of FLOSS projects on GitHub, download all the comments for those projects using GitHub REST API, and conduct an empirical study. With SGID4SE, such an empirical study would require less manual labeling than a keyword-based approach. Finally, duplicate instances in a dataset can provide inflated results if the same text belongs in both the training and test partitions. To avoid such pitfalls, we ensured unique instances during dataset curation. Therefore, our results do not suffer from such a threat to validity. **External validity:** Selection of datasets from FLOSS projects hosted on GitHub imposes an internal threat to our study. We have curated GitHub, which does not represent the many FLOSS projects. FLOSS projects vary widely based on governance model, CoC guidelines, CoC enforcement mechanisms, and above all, community values. Therefore, this study replicated on a different FLOSS project such as Android, Chromium OS, Wikimedia, or Mozilla are likely to identify distributions among the SGID categories.

8 Conclusion

Prior studies prove the manifestation of SGID content that results in push-back of women's participation in FLOSS communities [77, 78]. To the best of our knowledge, we conduct the first study that attempts to automatically identify SGID content from software developers' interactions. In this regard, we have built a labeled corpus of 11,007 pull request comments for SGID identification and developed SGID4SE that achieves 95.7% accuracy with 82.8% f1-score. We released our labeled dataset, pre-trained models, results, and summary of error analysis in the replication package. To lessen SGID content in FLOSS communities and create a more inclusive environment, administrators can adopt our dataset and tool, enhance it further, and implement it for automated identifications.

Data availability

We have made our labeled dataset, source code, and results publicly available on GitHub at: https://github.com/WSU-SEAL/SGID4SE

References

[1] [n. d.]. Hatebase dataset. https://hatebase.org/. [Online; accessed on December 23, 2022].

[2] [n. d.]. LGBT slang. https://en.wikipedia.org/wiki/LGBT_slang. [Online; accessed on December 23, 2022].

[3] [n. d.]. Pejorative terms for women. https://en.wikipedia.org/wiki/Category:Pejorative_terms_for_women. [Online; accessed on December 23, 2022].

[4] [n. d.]. Search: Github docs. https://docs.github.com/en/rest/search. [Online; accessed on December 23, 2022].

[5] Content Consumer. 2008. [n. d.]. The great Ubuntu-girlfriend experiment. http://contentconsumer.wordpress.com/2008/04/27/is-ubuntu-useableenough-for-my-girlfriend/. [Online; accessed June 01, 2014].

[6] Resham Ahluwalia, Evgeniia Shcherbinina, Edward Callow, Anderson CA Nascimento, and Martine De Cock. 2018. Detecting Misogynous Tweets.. In *IberEval@ SEPLN*. 242–248.

[7] Toufique Ahmed, Amiangshu Bosu, Anindya Iqbal, and Shahram Rahimi. [n. d.]. SentiCR: a customized sentiment analysis tool for code review interactions. In *2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 106–111.

[8] Conversation AI. [n. d.]. What if technology could help improve conversations online? https://www.perspectiveapi.com/

[9] Conversation AI. 2018. Annotation instructions for Toxicity with sub-attributes. https://github.com/conversationai/conversationai.github.io/blob/master /crowdsourcing_annotation_schemes/toxicity_with_subattributes.md.

[10] Anonymous. 2014. Leaving Toxic Open Source Communities. https://modelviewculture.com/pieces/leaving-toxic-open-source-communities

[11] Maria Anzovino, Elisabetta Fersini, and Paolo Rosso. 2018. Automatic identification and classification of misogynistic language on twitter. In *International Conference on Applications of Natural Language to Information Systems*. Springer, 57–64.

[12] Catherine Ashcraft, Brad McLain, and Elizabeth Eger. 2016. *Women in tech: The facts.* National Center for Women & Technology (NCWIT) Colorado, CO, USA.

[13] Ioana Baldini, Dennis Wei, Karthikeyan Natesan Ramamurthy, Moninder Singh, and Mikhail Yurochkin. 2022. Your fairness may vary: Pretrained language model fairness in toxic text classification. In *Findings of the Association for*

Computational Linguistics: ACL 2022. 2245–2262.

[14] Valerio Basile, Maria Di Maro, D Croce, and L Passaro. 2020. Evalita 2020: Overview of the 7th evaluation campaign of natural language processing and speech tools for italian. In *7th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop, EVALITA 2020*, Vol. 2765. CEUR-ws.

[15] Erin Baucom. 2018. An exploration into archival descriptions of LGBTQ materials. *The American Archivist* 81, 1 (2018), 65–83.

[16] Shiladitya Bhattacharya, Siddharth Singh, Ritesh Kumar, Akanksha Bansal, Akash Bhagat, Yogesh Dawer, Bornini Lahiri, and Atul Ojha. 2020. Developing a Multilingual Annotated Corpus of Misogyny and Aggression.

[17] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the association for computational linguistics* 5 (2017), 135–146.

[18] Jonathon Byrd and Zachary Lipton. 2019. What is the effect of importance weighting in deep learning?. In *International conference on machine learning*. PMLR, 872–881.

[19] Jose Sebastián Canós. 2018. Misogyny Identification Through SVM at IberEval 2018.. In *Ibereval@ sepln*. 229–233.

[20] Kajohnsak Chaokromthong, Nittaya Sintao, et al. 2021. Sample Size Estimation using Yamane and Cochran and Krejcie and Morgan and Green Formulas and Cohen Statistical Power Analysis by G* Power and Comparisions. *Apheit International Journal* 10, 2 (2021), 76–86.

[21] Davide Chicco and Giuseppe Jurman. 2020. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC genomics* 21, 1 (2020), 1–13.

[22] Patricia Chiril, Farah Benamara, and Véronique Moriceau. 2021. "Be nice to your wife! The restaurants are closed": Can Gender Stereotype Detection Improve Sexism Classification?. In *Findings of the Association for Computational Linguistics: EMNLP 2021*. 2833–2844.

[23] Lorraine Code. 2002. *Encyclopedia of feminist theories*. Routledge.

[24] Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement* 20, 1 (1960), 37–46.

[25] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[26] L. DiDio. [n. d.]. Look Out for Techno-Hazing. 31, 39 ([n. d.]), 72–73.

[27] Melanie Ehrenkranz. 2017. Women engineers get real about the worst sexism they've experienced at work. https://www.mic.com/articles/181968/women-engineers-get-real-about-the-worst-sexism-theyve-experienced-at-work.

[28] Nelly Elsayed, Anthony S Maida, and Magdy Bayoumi. 2018. Deep gated recurrent and convolutional network hybrid model for univariate time series classification. *arXiv preprint arXiv:1812.07683* (2018).

[29] Tracie Farrell, Miriam Fernandez, Jakub Novotny, and Harith Alani. 2019. Exploring Misogyny across the Manosphere in Reddit *(WebSci '19)*. Association for Computing Machinery, New York, NY, USA, 87–96. doi:10.1145/3292522.3326045

[30] Diane Felmlee, Paulina Inara Rodis, and Amy Zhang. 2020. Sexist slurs: Reinforcing feminine stereotypes online. *Sex Roles* 83, 1 (2020), 16–28.

[31] Isabella Ferreira, Jinghui Cheng, and Bram Adams. 2021. The" shut the f** k up" phenomenon: Characterizing incivility in open source code review discussions. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW2 (2021), 1–35.

[32] Isabella Ferreira, Ahlaam Rafiq, and Jinghui Cheng. 2024. Incivility detection in open source code review and issue discussions. *Journal of Systems and Software* 209 (2024), 111935.

[33] Elisabetta Fersini, Paolo Rosso, and Maria Anzovino. 2018. Overview of the Task on Automatic Misogyny Identification at IberEval 2018. *Ibereval@ sepln* 2150 (2018), 214–228.

[34] Simona Frenda, Ghanem Bilal, et al. 2018. Exploration of Misogyny in Spanish and English tweets. In *Third workshop on evaluation of human language technologies for iberian languages (ibereval 2018)*, Vol. 2150. Ceur Workshop Proceedings, 260–267.

[35] Carolina Pía García Johnson and Kathleen Otto. 2019. Better together: A model for women and LGBTQ equality in the workplace. *Frontiers in Psychology* 10 (2019), 272.

[36] Iakes Goenaga, Aitziber Atutxa, Koldo Gojenola, Arantza Casillas, Arantza Díaz de Ilarraza, Nerea Ezeiza, Maite Oronoz, Alicia Pérez, and Olatz Perez-de Viñaspre. 2018. Automatic Misogyny Identification Using Neural Networks.. In *IberEval@ SEPLN*. 249–254.

[37] Georgios Gousios. 2013. The GHTorrent dataset and tool suite. In *Proceedings of the 10th Working Conference on Mining Software Repositories* (San Francisco, CA, USA) *(MSR '13)*. IEEE Press, Piscataway, NJ, USA, 233–236. http://dl.acm.org/citation.cfm?id=2487085.2487132

[38] Georgios Gousios, Martin Pinzger, and Arie van Deursen. 2014. An exploratory study of the pull-based software development model. In *Proceedings of the 36th international conference on software engineering*. 345–355.

[39] Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural networks* 18, 5-6 (2005), 602–610.

[40] Ella Guest, Bertie Vidgen, Alexandros Mittos, Nishanth Sastry, Gareth Tyson, and Helen Margetts. 2021. An expert annotated dataset for the detection of online misogyny. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. 1336–1350.

[41] Guo Haixiang, Li Yijing, Jennifer Shang, Gu Mingyun, Huang Yuanyue, and Gong Bing. 2017. Learning from class-imbalanced data: Review of methods and applications. *Expert systems with applications* 73 (2017), 220–239.

[42] Paul B Hanton. 2015. The lack of women in technology: The role culture and sexism play. (2015).

[43] Sarah Hewitt, T. Tiropanis, and C. Bokhove. 2016. The Problem of Identifying Misogynist Language on Twitter (and Other Online Social Spaces). In *Proceedings of the 8th ACM Conference on Web Science* (Hannover, Germany) *(WebSci '16)*. Association for Computing Machinery, New York, NY, USA, 333–335. doi:10.1145/2908131.2908183

[44] Emma Alice Jane. 2014. 'Back to the kitchen, cunt': Speaking the unspeakable about online misogyny. *Continuum* 28, 4 (2014), 558–570.

[45] Akshita Jha and Radhika Mamidi. 2017. When does a compliment become sexist? Analysis and classification of ambivalent sexism using twitter data. In *Proceedings of the Second Workshop on NLP and Computational Social Science*. Association for Computational Linguistics, Vancouver, Canada, 7–16. doi:10.18653/v1/W17-2902

[46] Rie Johnson and Tong Zhang. 2017. Deep pyramid convolutional neural networks for text categorization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 562–570.

[47] Robbert Jongeling, Proshanta Sarkar, Subhajit Datta, and Alexander Serebrenik. 2017. On negative results when using sentiment analysis tools for software engineering research. *Empirical Software Engineering* 22, 5 (2017), 2543–2584.

[48] Cheris Kramarae and Dale Spender. 2004. *Routledge international encyclopedia of women: Global women's issues and knowledge*. Routledge.

[49] Oliver Kramer and Oliver Kramer. 2016. Scikit-learn. *Machine learning for evolution strategies* (2016), 45–53.

[50] Klaus Krippendorff. 2011. Computing Krippendorff's alpha-reliability.

[51] Ritesh Kumar, Atul Kr. Ojha, Shervin Malmasi, and Marcos Zampieri. 2018. Benchmarking Aggression Identification in Social Media. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*. Association for Computational Linguistics, Santa Fe, New Mexico, USA, 1–11.

[52] Keita Kurita, Anna Belova, and Antonios Anastasopoulos. 2020. Towards Robust Toxic Content Classification. arXiv:1912.06872.

[53] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942* (2019).

[54] Han Liu, Fatima Chiroma, and Mihaela Cocea. 2018. Identification and Classification of Misogynous Tweets Using Multi-classifier Fusion.. In *Ibereval@ sepln*. 268–273.

[55] Adrienne Massanari. 2017. # Gamergate and The Fappening: How Reddit's algorithm, governance, and culture support toxic technocultures. *New media & society* 19, 3 (2017), 329–346.

[56] Courtney Miller, Sophie Cohen, Daniel Klug, Bogdan Vasilescu, and Christian KaUstner. 2022. " Did you miss my comment or what?" understanding toxicity in open source discussions. In *Proceedings of the 44th International Conference on Software Engineering*. 710–722.

[57] MSang. [n. d.]. HatEval dataset. https://github.com/msang/hateval/blob/master/keyword_set.md. [Online; accessed on April 23, 2022].

[58] Hala Mulki and Bilal Ghanem. 2021. Let-Mi: An Arabic Levantine Twitter Dataset for Misogynistic Language.

[59] Arianna Muti and Alberto Barron-Cede. 2020. UniBO@ AMI: A Multi-Class Approach to Misogyny and Aggressiveness Identification on Twitter Posts Using AlBERTo. *EVALITA Evaluation of NLP and Speech Tools for Italian-December 17th, 2020* (2020), 29.

[60] Victor Nina-Alcocer. 2018. AMI at IberEval2018 Automatic Misogyny Identification in Spanish and English Tweets.. In *Ibereval@ sepln*. 274–279.

[61] Stack Overflow. 2023. 2023 Developer Survey. https://survey.stackoverflow.co/2022/.

[62] Endang Wahyu Pamungkas, Valerio Basile, and Viviana Patti. 2020. Misogyny detection in twitter: a multilingual and cross-domain study. *Information Processing & Management* 57, 6 (2020), 102360.

[63] Endang Wahyu Pamungkas, Alessandra Teresa Cignarella, Valerio Basile, Viviana Patti, et al. 2018. 14-ExLab@ UniTo for AMI at IberEval2018: Exploiting lexical knowledge for detecting misogyny in English and Spanish tweets. In *3rd Workshop on Evaluation of Human Language Technologies for Iberian Languages, IberEval 2018*, Vol. 2150. CEUR-WS, 234–241.

[64] Rajshakhar Paul, Amiangshu Bosu, and Kazi Zakia Sultana. 2019. Expressions of Sentiments during Code Reviews: Male vs. Female. In *Proceedings of the 26th IEEE International Conference on Software Analysis, Evolution and Reengineering* (Hangzhou, China) *(SANER '19)*.

[65] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *the Journal of machine Learning research* 12 (2011), 2825–2830.

[66] Marco Polignano, Pierpaolo Basile, Marco De Gemmis, Giovanni Semeraro, and Valerio Basile. 2019. Alberto: Italian BERT language understanding model for NLP challenging tasks based on tweets. In *6th Italian Conference on Computational Linguistics, CLiC-it 2019*, Vol. 2481. CEUR, 1–6.

[67] Naveen Raman, Min Cao, Yulia Tsvetkov, Christian Kästner, and Bogdan Vasilescu. 2020. Stress and Burnout in Open Source: Toward Finding, Understanding, and Mitigating Unhealthy Interactions. *2020 IEEE/ACM 42nd International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER)* (2020), 57–60.

[68] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084* (2019).

[69] Francisco Rodríguez-Sánchez, Jorge Carrillo-de Albornoz, and Laura Plaza. 2020. Automatic classification of sexism in social networks: An empirical study on twitter data. *IEEE Access* 8 (2020), 219563–219576.

[70] Jaydeb Sarker, Sayma Sultana, Steve Wilson, and Amiangshu Bosu. 2023. ToxiSpanSE: An Explainable Toxicity Detection in Code Review Comments. In *Proceedings of the 17th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM).*

[71] Jaydeb Sarker, Asif Kamal Turzo, and Amiangshu Bosu. 2020. A benchmark study of the contemporary toxicity detectors on software engineering interactions. In *2020 27th Asia-Pacific Software Engineering Conference (APSEC).* IEEE, 218–227.

[72] Jaydeb Sarker, Asif Kamal Turzo, Ming Dong, and Amiangshu Bosu. 2023. Automated Identification of Toxic Code Reviews Using ToxiCR. *ACM Transactions on Software Engineering and Methodology* 32, 5, Article 118 (jul 2023), 32 pages. doi:10.1145/3583562

[73] Carl-Erik Särndal, Bengt Swensson, and Jan Wretman. 2003. *Model assisted survey sampling.* Springer Science & Business Media.

[74] Richard T Schaefer and Bonnie Haaland. 2011. *Sociology: A brief introduction.* McGraw-Hill New York.

[75] Roger Scruton. 2007. *The Palgrave Macmillan dictionary of political thought.* Springer.

[76] Elena Shushkevich and John Cardiff. 2018. Classifying Misogynistic Tweets Using a Blended Model: The AMI Shared Task in IBEREVAL 2018.. In *Ibereval@ sepln.* 255–259.

[77] Vandana Singh and Brice Bongiovanni. 2021. Motivated and Capable but No Space for Error. *The International Journal of Information, Diversity, & Inclusion* 5, 3 (2021), 98–126.

[78] Megan Squire and Rebecca Gazda. 2015. FLOSS as a Source for Profanity and Insults: Collecting the Data. In *2015 48th Hawaii International Conference on System Sciences.* 5290–5298. doi:10.1109/HICSS.2015.623

[79] Kalpana Srivastava, Suprakash Chaudhury, PS Bhat, and Samiksha Sahu. 2017. Misogyny, feminism, and sexual harassment. *Industrial psychiatry journal* 26, 2 (2017), 111.

[80] Sayma Sultana. 2022. Identifying Sexism and Misogyny in Pull Request Comments. In *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering.* 1–3.

[81] Sayma Sultana, London Ariel Cavaletto, and Amiangshu Bosu. 2021. Identifying the Prevalence of Gender Biases among the Computing Organizations. *arXiv preprint arXiv:2107.00212* (2021).

[82] Sayma Sultana, Jaydeb Sarker, and Amiangshu Bosu. 2021. A Rubric to Identify Misogynistic and Sexist Texts from Software Developer Communications *(ESEM '21).* Association for Computing Machinery, New York, NY, USA, Article 27, 6 pages. doi:10.1145/3475716.3484189

[83] Sayma Sultana, Jaydeb Sarker, and Amiangshu Bosu. 2021. *A Rubric to Identify Misogynistic and Sexist Texts from Software Developer Communications.* Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3475716.3484189

[84] Janet K Swim, Kathryn J Aikin, Wayne S Hall, and Barbara A Hunter. 1995. Sexism and racism: Old-fashioned and modern prejudices. *Journal of personality and social psychology* 68, 2 (1995), 199.

[85] 7 Steps to Learn English. 2022. Gender of Nouns: Useful Masculine and Feminine List. https://7esl.com/gender-of-nouns/.

[86] Bianca Trinkenreich, Ricardo Britto, Marco Aurelio Gerosa, and Igor Steinmacher. 2022. An Empirical Investigation on the Challenges Faced by Women in the Software Industry: A Case Study. *arXiv preprint arXiv:2203.10555* (2022).

[87] Trae Vassallo, Ellen Levy, Michele Madansky, Hillary Mickell, Bennett Porter, and Monica Leas. 2017. *Elephant in the Valley.* https://www.elephantinthevalley.com/

[88] Zeerak Waseem, Thomas Davidson, Dana Warmsley, and Ingmar Weber. 2017. Understanding Abuse: A Typology of Abusive Language Detection Subtasks. In *Proceedings of the First Workshop on Abusive Language Online.* Association for Computational Linguistics, Vancouver, BC, Canada, 78–84.

[89] Chris J Young. 2015. Understanding Games and the Industry that Produces Them: A Review of the Edited Volume The Video Game Industry. *Journal of Games Criticism* 2, 2 (2015), X–X.

[90] Philine Zeinert, Nanna Inie, and Leon Derczynski. 2021. Annotating online misogyny. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers).* 3181–3197.

[91] Ting Zhang, Bowen Xu, Ferdian Thung, Stefanus Agus Haryono, David Lo, and Lingxiao Jiang. 2020. Sentiment analysis for software engineering: How far can pre-trained transformer models go?. In *2020 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. IEEE, 70–80.

[92] Zhuoyuan Zheng, Yunpeng Cai, and Ye Li. 2015. Oversampling method for imbalanced classification. *Computing and Informatics* 34, 5 (2015), 1017–1037.