Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control
Journal homepage: http://kinetik.umm.ac.id
ISSN: 2503-2267
Vol. 8, No. 3, August, Pp. 651-662

Evaluation of stratified k-fold cross validation for predicting bug severity in game review classification

Mustika Kurnia Mayangsari*[1], Iwan Syarif[1], Aliridho Barakbah[1]
Politeknik Elektronika Negeri Surabaya, Indonesia[1]

Article Info
Keywords:
Steam, Game Review, Bug Severity, KNN, Decision Tree, Naïve Bayes, Text Classification, N-Gram, SKCV

Article history:
Received: May 24, 2023
Accepted: July 01, 2023
Published: August 31, 2023

Cite:
M. K. Mayangsari, I. Syarif, and A. Barakbah, "Evaluation of Stratified K-Fold Cross Validation for Predicting Bug Severity in Game Review Classification", KINETIK, vol. 8, no. 3, Aug. 2023.
https://doi.org/10.22219/kinetik.v8i3.1740

*Corresponding author.
Mustika Kurnia Mayangsari
E-mail address:
mustikakurniam@gmail.com

Abstract
Steam review data provides a lot of information for the game development team, either positive or negative reviews. It is essential as negative and positive reviews provide crucial information, and 7% of positive reviews contains bug reports. These bug reports were captured after the game was released, and many reports of common problems still exist. If players found an issue in the game, they could report it directly through the review feature provided by the online game platform. However, it took a long time for the development team to manually analyze and classify the reviews. This study proposed a new approach to automatically classify the reviews on Steam based on the bug severity level. Therefore, to solve this problem, we recommend a solution based on the research background indicated above. For this experiment, we analyzed reviews on two popular game titles namely, FIFA 23 and Apex Legends. We implemented three different classifiers, namely KNN, Decision Tree, and Naïve Bayes, which would be used to train a dataset to classify the bug severity level. Due to the imbalanced dataset, we performed cross-validation to reduce bias in the dataset. Performance in this model would be evaluated using accuracy rate, precision, recall, and F1 score. As a result, the experiment showed that game reviews of different game titles achieved different accuracy scores. The game review classification for FIFA 23 performed better than the game review classification for Apex Legends. The mean accuracy score of FIFA 23 was 72% with Decision Tree and Apex Legend was 64% with KNN.

1. Introduction

Since 2020, one of the most popular online game distribution platforms, namely Steam, game sales have continued to increase by more than 20 percent compared to sales in the previous year [1]. This increase continues to grow in 2021 as the number of downloads on the Steam mobile app reached 1.5 million times on iOS and Android [2]. Steam is a digital product similar to a mobile app store platform that allows users to sneak preview the games they have purchased and usually offers both paid and free games. Lin et al. conducted an empirical study by collecting game information data from the Steam Store related to game release info data and reviews from the Steam Community. The study makes an essential point for the authors: negative reviews provide information about game defects, and positive reviews provide crucial information. They found that out of all the positive reviews, 7% contained bug report information [3].

The game review has many essential benefits as a source of information for the game development team. After extracting data from game reviews, the indie game developer team will gain knowledge from the data to improve the gameplay or game features [4]. During game development, a game review is more subjective and vital as it measures game quality assurance because it can affect maintaining and determining the tools to implement in the next game development version [5]. Since the game industry's biggest challenge is developing successful games while keeping quality in mind, some issues still need to be fixed. Despite after release of the game to the market, many reports of frequent incidents, such as unintentional or unexpected behavior, still exist. Program code errors cause this because the development team did not anticipate certain cases resulting in unexpected behavior. Currently, if players find an issue in the game, they can report it directly through the review feature provided by the online game platform.

However, this is proven by the decrease in player count caused by bug reports in game reviews. We attempted to conduct a case study on Apex Legends game at the beginning of Q4 in 2022 and noticed a decrease in the number of players which fluctuated insignificantly [6]. The current situation is that many bug-related info in game reviews still occurs and is ignored by the development team. The problem is the development team takes a long time to analyze and categorize the game review. This study will propose a new approach to automatically solve the problem of categorizing game reviews on Steam based on the bug severity level. Furthermore, referring to Levy and Novak classified the severity of bugs in the game into four classes: low, medium, high, and critical bugs [7]. Determining the

nomenclature of bug severity might be subjective from developer to developer. Lamkanfi et al. studied the bug severity of three different open-source applications. They did not include normal bug severity levels because they were in a gray zone and could get confusing during the classification process. Because medium severity intersects low and high severity, we will omit the medium severity [8]. After simplifying the bug severity levels in game reviews, the severity levels are divided into low, high, and critical levels.

Many previous studies have researched the classification of user reviews and applied various methods. Maalej and Nabil created probabilistic methodologies and heuristics for categorizing reviews based on metadata (for example, star rating and text length), keyword frequencies, linguistic rules, and sentiment analysis [9]. They transformed user reviews into Bag-of-Word (BoW) format and then classified them using three classification techniques: Naive Bayes, Decision Tree, and MaxEnt. Parwita and Siahaan performed the classification method twice, with the first classifier classifying the informative reviews and the second classifier classifying the bug and feature request categories [10]. However, they still discovered issues that could impact the overall performance. The problems are the effect of sentence length on the dimension of the word vector [10], slang words [11], and writing incorrect grammar in review sentences causing performance changes [11]. Besides that, prior research only focused on mobile application reviews in the Google Play, Apple App Stores, and web-based bug tracking systems [8]–[12].

Therefore, in this study, this study concentrates on game reviews on a specialized online platform for game distribution with a more specific user niche. We suggest introducing a solution based on the research background indicated above, where three different classifiers will be used to train a dataset to classify the bug severity level using cross-validation. Performance in this model will be evaluated using accuracy rate, precision, recall, and F1 score. To increase the precision of this classification model, we extract features using the TF-IDF vectorizer, then use a threshold to select important features. Additionally, we used specific analogies or comparisons to evaluate the performance of three classifiers.

2. Research Method

This research involves the following phases: 1) data collecting; 2) text pre-processing; 3) TF-IDF and keyword filtering; 4) splitting data using stratified k-fold cross validation; 5) constructing a classification model using KNN, Decision Tree, and Naïve Bayes; 5) evaluating performance. The detail of the overall system process can be seen in Figure 1 below:

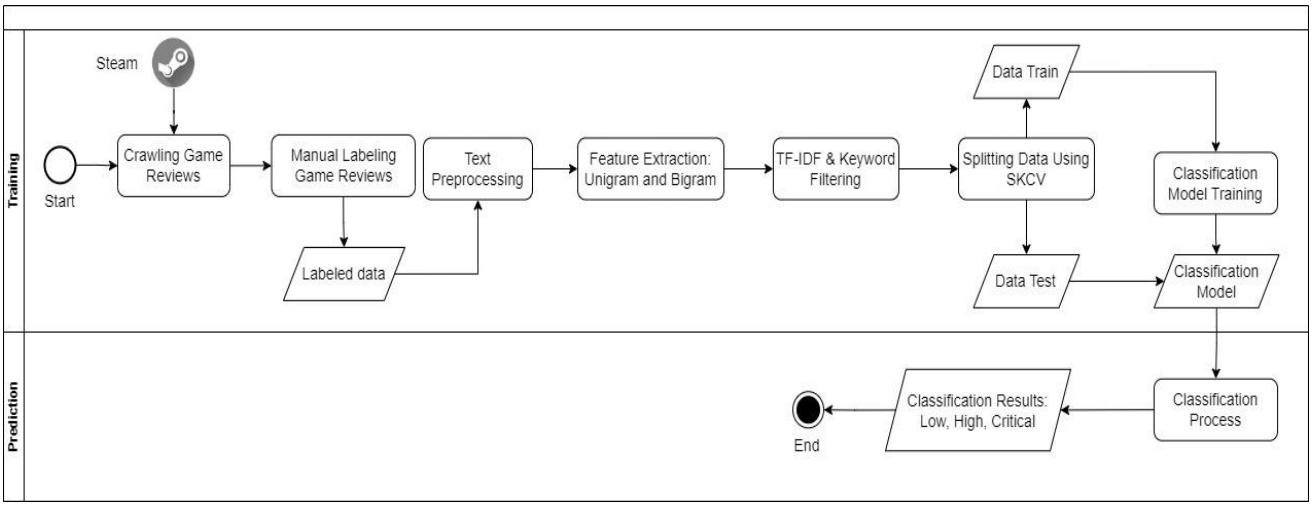

Figure 1. System design

2.1 Data Collecting

In this study, we crawled game reviews in Steam using an existing crawler[1] to collect experimental data. We extracted positive and negative reviews because both discuss game defects. The game reviews were sourced from FIFA 23 and Apex Legends, as both games were most played among users. Furthermore, the data required for the classification process later was the review text and the length of review sentences. The data obtained is filtered based on the English language and amounted to the last reviews in Q4 in 2022 and Q1 in 2023.

At the stage of dataset design, manual review classification is performed based on the predetermined classes, namely, unknown, low, high, and critical. This manual classification was performed by three participants with expertise

[1] https://github.com/aesuli/steam-crawler

in QA or game testing and familiarity with Steam reviews. As they labeled each review text based on predetermined categories, each category has an explanation and description that can be seen in Table 1.

Table 1. Game review labels description

Labels	Description
Low	A bug report that does not really have an impact on the gameplay
High	A bug report that has a big impact on the game experience
Critical	A bug report that bothers players so much that they can't play or enjoy the game.
Unknown	The review explains the bug report but does not describe how the player experienced the bug

2.2 Text Pre-processing

Before classifying the review dataset, it is necessary to perform text preprocessing to make it easier for the algorithm to classify the text. These are the steps involved in text preprocessing:

2.2.1 Case Folding

The first step is case folding. It is the process of transforming words into the same forms, either lowercase or uppercase [13]. In this research, we preferably transformed the review text into lowercase since it could reduce problems that lead to information loss. An example of converting review text into lowercase is shown in Table 2:

Table 2. Lower casing

Original Review	Result
I can confidently say I wasted money this is honestly the worst game I've ever played... After a hundred hours of raging because of bugs and glitches, I uninstalled it. The amount of times that my character in career mode has been completely fouled and nothing was called is insane.	i can confidently say i wasted money this is honestly the worst game i've ever played... after a hundred hours of raging because of bugs and glitches, i uninstalled it. the amount of times that my character in career mode has been completely fouled and nothing was called is insane.

2.2.2 Text Standardization

After translating non-English reviews, the following step is to standardize the text by expanding contractions. The purpose of expanding contractions for reducing vocabulary size by transforming contractions to complete phrases for consistency [14]. The following example of expanding contraction is in Table 3:

Table 3. Expand contraction text

Original Review	Result
i havent been able to play the game for weeks i've tried everything i get stuck on the ea anticheat logo and it gives me the "game could not start, administrator access required"	i have not been able to play the game for weeks i have tried everything i get stuck on the ea anticheat logo and it gives me the "game could not start, administrator access required"

2.2.3 Handling Slang Words

The following process is to handle the informal English words or slangs frequently used in daily conversations, social media, and SMS communication. Nevertheless, due to the necessity of researching slang words in their original form, we scraped one of the slang dictionary websites (https://noslang.com) to reduce ambiguity and refine the meaning of game review analysis [15]. Then, we restored it as a JSON dictionary file and constructed a slang dictionary from the game review sentences. After building a slang dictionary, we replaced slang words with original words as shown in Table 4:

Table 4. Replacing slang words

Original Review	Result
i cannot play the game bcs of the anticheat! should i request a refund or ea will fix it these days?	i cannot play the game because of the anticheat! should i request a refund or ea will fix it these days?

2.2.4 Data Cleaning

Some of the game reviews contain URLs. We need to handle the URLs contained in the game review because it has no meaning and cannot provide additional information for analyzing the sentence [16]. In any case, according to text pre-processing research by Işık and Dag that URLs can be useful for providing text-related information that is hard to obtain from the context of certain applications [16]. Furthermore, the game review also contains non-ASCII characters, numbers, and punctuation. Thus, we remove them by using regex or regular expression syntax to handle nonessential characters in the game review as shown in Table 5:

Table 5. Data Cleaning

Original Review	Result
almost 500 hours of playtime....i am going out of my way to tell respawn.....fix your ã¢â„¢ã¢â„¢ã¢â„¢ã¢â„¢ã¢â„¢ã¢â„¢ servers mr. developer man!!!! this is like pubg on xbox early release bad. edit; season 11, servers are working as they should, no issues!! thank god....and ash got the gyroscopic thick robo hips though!!!!	almost hours of playtime i am going out of my way to tell respawn fix your servers mr developer man this is like pubg on xbox early release bad edit season servers are working as they should no issues thank god and ash got the gyroscopic thick robo hips though

2.2.5 Tokenization

Tokenization, the next step is to continue with the tokenization process by separating the cleaned review texts where there are spaces into meaningful word chunks. We can see the tokenization results in Table 6:

Table 6. Tokenization results

Original Review	Result
almost hours of playtime i am going out of my way to tell respawn fix your servers mr developer man this is like pubg on xbox early release bad edit season servers are working as they should no issues thank god and ash got the gyroscopic thick robo hips though	almost, hours, of, playtime, i, am, going, out, of, my, way, to, tell, respawn, fix, your, servers, mr, developer, man, this, is, like, pubg, on, xbox, early, release, bad, edit, season, servers, are, working, as, they, should, no, issues, thank, god, and, ash, got, the, gyroscopic, thick, robo, hips, though

2.2.6 Removing Stop Words

Stop words are common words that appear frequently but have nonessential meanings and eliminate them to find important keywords in a game review. In addition, this study used the NLTK (Natural Language Toolkit) library provided by Python to remove stop words. Additionally, we also added a few words in the list of custom stop words to ignore them in the review text. The following Table 7 is the result of stop words removal:

Table 7. Stop words removal

Original Review	Result
almost, hours, of, playtime, i, am, going, out, of, my, way, to, tell,	hours, playtime, going, way, tell, respawn, fix, servers, developer,

respawn, fix, your, servers, mr, developer, man, this, is, like, pubg, on, xbox, early, release, bad, edit, season, servers, are, working, as, they, should, no, issues, thank, god, and, ash, got, the, gyroscopic, thick, robo, hips, though	early, release, bad, edit, season, servers, working, issues, thank, god, ash, got, gyroscopic, thick, robo, hips, though

2.2.7 Lemmatization

Lemmatization is the final pre-processing process. It is similar to stemming in removing affixes to arrive at the word's root form. However, this form is also called the root word or lemma, not the root stem. The robust lemmatization module, the NLTK package, uses WordNet and the word's syntax and semantics, such as part of speech and context, to standardize the review text [17]. The following is an example of lemmatizing words into root words as shown in Table 8:

Table 8. Lemmatization results

Original Review	Result
hours playtime going way tell respawn fix servers developer early release bad edit season servers working issues thank god ash got gyroscopic thick robo hips though	hour playtime go way tell respawn fix server developer early release bad edit season server work issue thank god ash get gyroscopic thick robo hips though

2.3 Feature Extraction

2.3.1 TF-IDF

After cleaning the data with text preprocessing, we built the Bag of Words (BoW) model to transform text documents into vectors. Each document is converted into a vector representing the frequency of all distinct words present in that document's vector space. Subsequently, we applied the term weighting method to rank all texts in the vector space model by calculating the weight based on the statistical information of a term in the document. Fundamentally, this form of weighting consists of two elements. The first component, TF, indicates the frequency of the term in a document [18]. The second component is the inverse document frequency (IDF), which indicates that a term that appears in multiple documents should be assigned a low weight or the IDF value decreases [19]. The following formula of Equation 1 computes the TF value, whereas Equation 2 calculates the IDF value. The calculation of word frequency scores that highlight interesting words shows in Equation 3.

$$TF(t,d) = \frac{number\ of\ times\ t\ appears\ in\ d}{total\ number\ of\ terms\ in\ d} \quad (1)$$

$$IDF(t) = log\frac{N}{1+df} \quad (2)$$

$$F_d = TF(t,d) * IDF(t) \quad (3)$$

Where:

t = represent of terms in a sentence
d = the frequency term (t) in the document
TF = term frequency
IDF = inverse document frequency
F_d = term frequency matrix

2.3.2 Keyword Filtering

The following step is to filter the data using several related keywords. We selected keywords by filtering the term frequency. We set the filtering threshold at 50% of the maximum term frequency as shown in Equation 4 [20]:

$$T_d = \frac{1}{2}\ max\ \{F_d\} \quad (4)$$

To reduce the number of terms, we apply a filtering threshold to the term frequency, and the remaining terms are referred to as new keywords shown in Equation 5 [20]:

$$F'_d = \{ f_{ti^d} \mid f_{ti^d} \geq T_d \}, t \in d \quad (5)$$

Where:
T_d = threshold of keywords filtering
F'_d = keywords filtering results
f = represent of keywords

2.4 Stratified K-Fold Cross Validation

After finishing the keyword filtering process, we split the data using Stratified K-Fold Cross Validation (SKCV). SKCV is an easy method to implement that can prevent data duplication for each category, thus reducing bias in a data set [21]. By implementing stratified in this study, the number of feature proportions in the training and testing data will be the same as the original data [22][23]. Therefore, the data set for each bug severity class will not be randomly distributed into k-folds without interfering with the sample distribution ratio across classes. The following Figure 2 below is an illustration of the application of SKCV in this study:

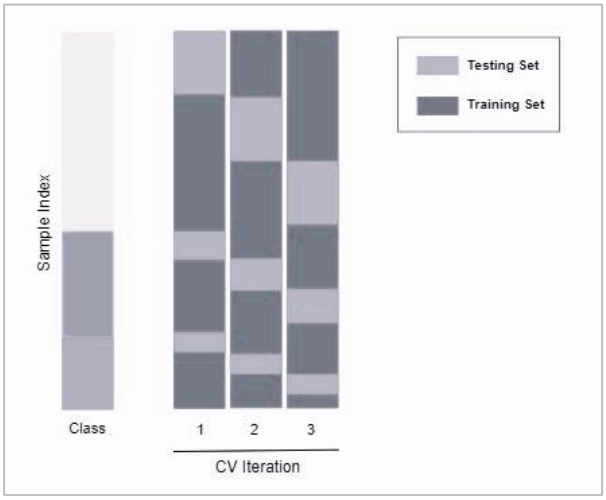

Figure 2. Stratified K-Fold Cross Validation (SKCV)

2.5 Classification Algorithms

Comparing three kinds of classification algorithms namely, K-Nearest Neighbor (KNN), Decision Tree, and Naïve Bayes, to determine which one is the best algorithm for game review text. We used KNN because it is one of the simple and effective methods for text categorization [24]. Then, we also applied the Decision Tree algorithm in the game review classification because it is the most robust technique that is frequently used in various domains, especially for text classification [25]. Moreover, we implemented Naïve Bayes classifier since it is extensively used for text classification based on the conditional probability of selected features through feature selection [26].

2.5.1 Performance Evaluation

Model evaluation is an important step to select the best model among the three classification models. This study will use the accuracy score and confusion matrix as performance measurements. From the confusion matrix, will get the calculation value of TP (True Positive), FP (False Positive), TN (True Negative), and FN (False Negative). Then, we can calculate the accuracy rate, F1 score, precision, and recall measurements.

Firstly, we calculated the accuracy rate to calculate the percentage accuracy value of a classification algorithm. Here is Equation 6 to find the accuracy value of a classifier.

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \quad (6)$$

Secondly, we calculated the precision and recall. Precision is used to validate the classification algorithm and determine whether the classification value of “True” corresponds to actual data. Then, recall is also known as True

Positive Rate (TPR) and measures the frequency of correctly recognized positive samples [27]. The precision and recall were calculated by using Equation 7 and Equation 8.

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

Lastly, we calculated the F1 Score to find the weighted average of the precision and recall values. The F1 score were calculated by using Equation 9:

$$F1\ Score = 2 \times \frac{recall * precision}{recall + precision} \tag{9}$$

3. Results and Discussion

Most game reviews on Steam have the same topic of discussion among active users. In this study, we collected game review datasets from two game titles: Apex Legends and FIFA 23. When labeling the game reviews collectively, we found that the percentage of the *unknown* category was more dominant than the other categories. Based on our observation, game review sentences in *unknown* categories are praise or user suggestions for future game improvement. We dropped the data to avoid biased sentences during the classification process. After dropping the data labeled *unknown* and missing values, the total data for Apex Legends was 78 reviews and FIFA 23 was 683 reviews, which had a distribution as shown in Table 9.

Table 9. Class distribution

Class	Game Title	
	Apex Legends	FIFA 23
Low	10	66
High	29	188
Critical	39	429

3.1 N-Gram Analysis

Each game review possesses a distinct review subject. We studied the topic of game reviews using a graphical model called n-gram. We built vocabulary and created a dictionary to map each n-gram to a unique index. Then, the analysis of the n-gram frequent words presented shows that the following variables are connected to the model created in the research. Further analysis of the top 20 words frequency in unigrams also allows the identification of variables as presented in Figure 3 for the game FIFA 23 and Figure 4 for the game Apex Legends:

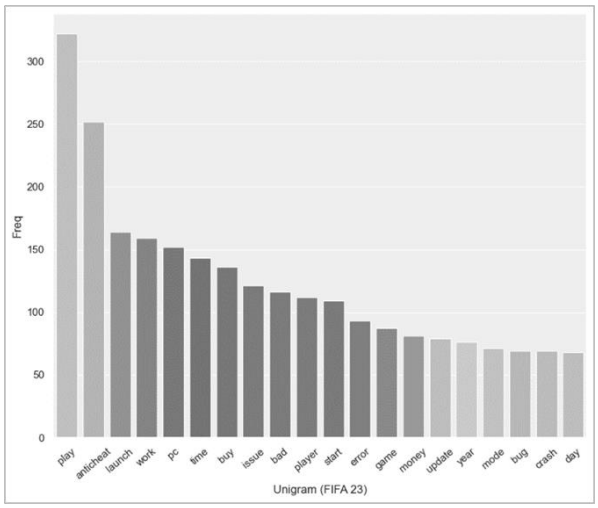

Figure 3. Unigram's analysis of FIFA 23

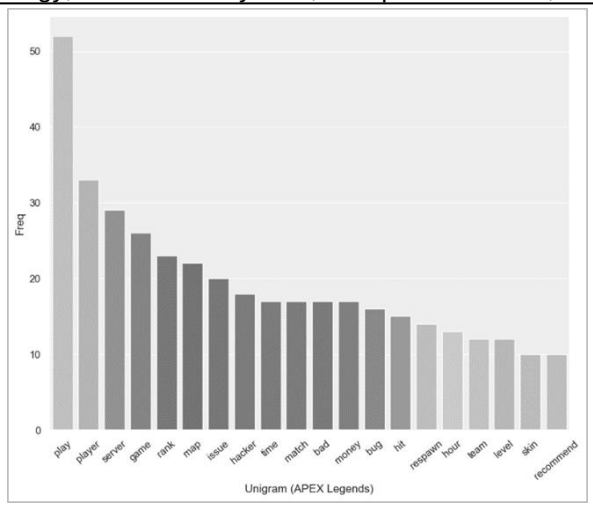

Figure 4. Unigram's analysis of Apex Legends

Additional information is provided by Bigram's analysis of the top 20 words' frequency. The motivating approaches covered in this study are those that relate to the bigram phrases as presented in Figure 5 for the game FIFA 23, while Figure shows the bigram phrases of the game Apex Legends.

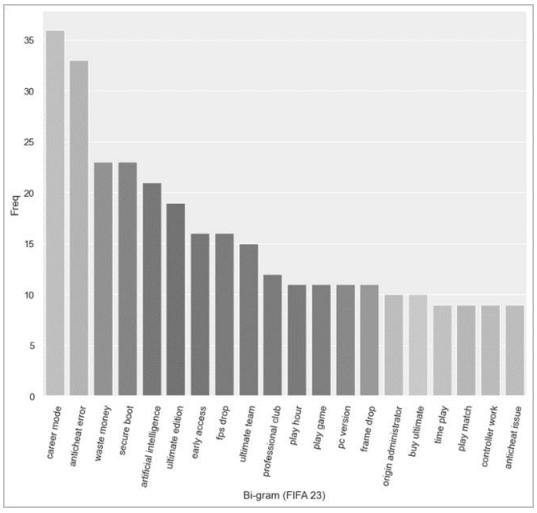

Figure 5. Bigram's analysis of FIFA 23

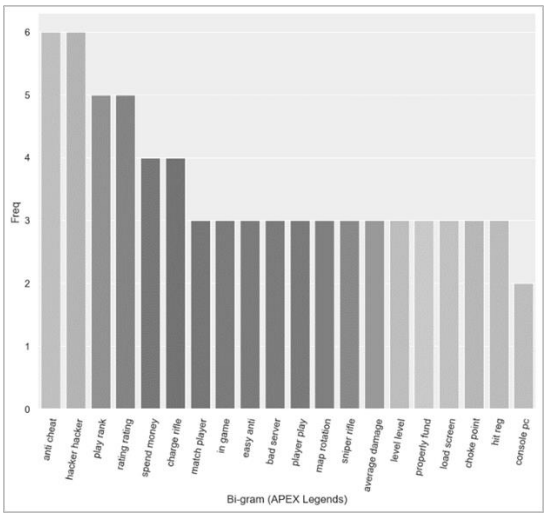

Figure 6. Bigram's analysis of Apex Legends

We applied the unigram and bigram n-gram language models in this study case. Numerous studies show that compared to bigram and trigram, unigram successfully achieves the highest accuracy in both English and non-English phrases [28]. However, Figure 3 and Figure 4 show that unigram still does not have the traits or characteristics of a bug report. Therefore, we selected multiple n-gram ranges by combining unequal n-grams model unigram and bigram, since the n-gram model combination provides a good accuracy [29]. As a result, the unigram's n-gram analysis reveals that the word "play" frequently appears. However, the term "play" is still irrelevant to accurately describe the entire analysis. After attempting to apply the bigram model, there are two syllables that can accurately describe the entire analysis. For example, in Figure 5 syllable term “career mode” frequently appears in FIFA 23 review, and in Figure 6 the syllable term “anti-cheat” frequently appears in the Apex Legends review.

According to the analysis of the n-gram results, the FIFA 23 game reports more on career mode problems. The meaning of the career mode problems in FIFA 23 is to cause several players to become stuck on the "Ready to Shine" screen after the career match starts [30]. Meanwhile, Apex Legends reports further information about anti-cheat and hacker concerns. Furthermore, despite Apex Legends' efforts to handle anti-cheats, hackers have been targeting it for the last few months and the situation is only getting worse [31].

3.2 Evaluating Classification Performance

In this process, we compared the outcomes of the different algorithms: KNN, Decision Tree, and Naïve Bayes. This study focused on several classification metrics: accuracy score, F1 score, precision, and recall. Besides that, CV is frequently used in machine learning to evaluate how well a model performs on untrained data. Due to the data imbalance as shown in Table 9, we implemented data splitting using SKCV to make the training and testing data more evenly distributed with the value of cross-validation n_split is 3. The percentage distribution of training and testing data by SKCV implementation is shown in Table 10. The following Table 11 compares the outcomes of the two review classification results with SKCV.

Table 10. The distribution of data training and testing

k	Apex Legends		FIFA 23	
	Training	Testing	Training	Testing
1	66.23%	33.77%	66.67%	33.33%
2	66.23%	33.77%	66.67%	33.33%
3	67.53%	32.47%	66.67%	33.33%

Table 11. Performance comparison of three classification methods

Metrics	Apex Legends			FIFA 23		
	KNN	Decision Tree	Naïve Bayes	KNN	Decision Tree	Naïve Bayes
Precision	64%	62%	50%	59%	70%	57%
Recall	64%	62%	49%	64%	72%	50%
F1-Score	62%	60%	50%	52%	71%	53%
Accuracy	64%	62%	49%	64%	72%	50%

The results of the classification experiments above with three algorithms: KNN, Decision Tree, and Naïve Bayes shown in the table above are the mean of classification evaluation results with stratified k-fold cross-validation (SKCV). For bug severity classification on Apex Legends game reviews, the highest mean accuracy of 64% was achieved when performing with the KNN. Additionally, the FIFA 23 game reviews performed better than the Apex Legends game reviews and achieved the highest mean accuracy of 72% when performing with the Decision Tree. However, Table 11 above shows that Naïve Bayes provides the lowest precision and recall results for both game reviews. It means that the Naïve Bayes method is still less precise when classifying the entire class well based on severity level.

The best accuracy results above are obtained from the confusion matrix as shown in Figure 7 using KNN and Figure 8 using Decision Tree. From the confusion matrix, we can gain the TP, TN, FP, and FN values. Multiclass classification is unlike binary classification since there are no positive or negative classes. Consequently, we need to find TP, TN, FP, and FN for each individual class. For example, we take the low class of Apex Legends, and Table 12 below are the metric values of the confusion matrix for the low class:

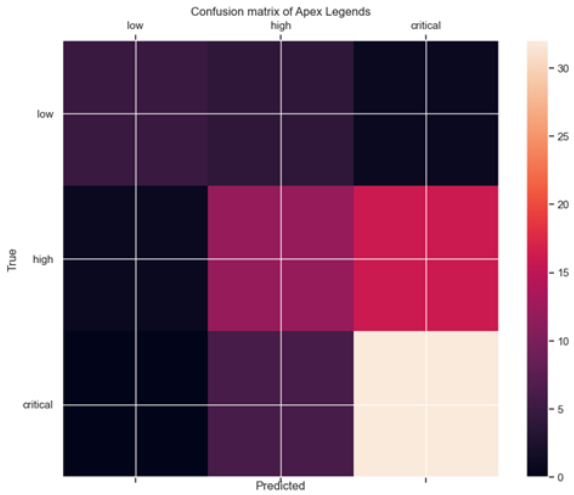

Figure 7. Confusion matrix of Apex Legends

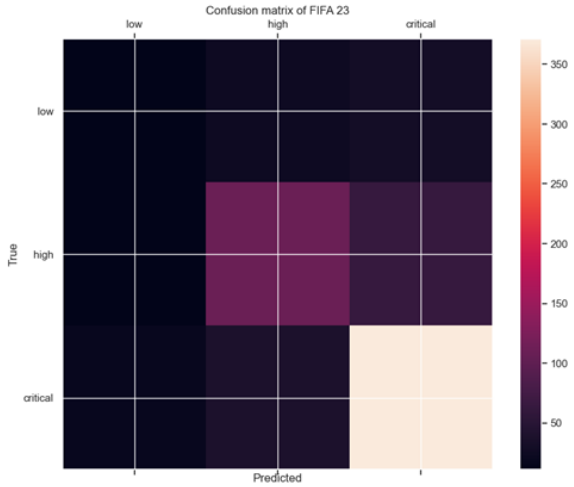

Figure 8. Confusion matrix of FIFA 23

Table 12. Example of confusion matrix calculation

Apex Legends	
TP	5
TN	66
FP	5
FN	6

4. Conclusion

Based on the studies that have been conducted, the following n-gram analysis using bigram revealed that the Apex Legends review during Q4 in 2022 frequently addressed issues regarding anti-cheat. While FIFA 23 players were primarily concerned with operating career mode issues during Q4 in 2022. In addition, SKCV performs well in terms of predicting the bug severity level. This is demonstrated by the fact that three classification algorithms produce unequal distribution of classes. According to our analytics, the game review classification for FIFA 23 performs better than the game review classification for Apex Legends, as shown in Table 11. However, Naïve Bayes had the lowest accuracy score for both game review classifications.

In future work, we suggest considering other alternative methods for feature extraction. This is due to the imbalance in the amount of class data and the ratio number is quite far. In addition, for future research, we recommend to propose a vector space-based method for extracting features that improve accuracy and reduce computation time. We will not implement all comparative features but filter specific features using context recognition. It should be reevaluated during the keyword feature extraction process in order to extract clustering-appropriate features.

Acknowledgment

We would like to thank my QA co-workers of the Gamification Team at Shopee Indonesia for their invaluable help and contributions throughout the research process. Their insights and knowledge were crucial in determining the direction of this research. Hopefully, this research can significantly improve the creative economy sector, especially applied to several software and gaming technology companies in Indonesia.

References

[1] Steamworks Development, "Steam - 2020 Year in Review.".
[2] B. Dean, "Steam Usage and Catalog Stats for 2022.".
[3] D. Lin, C.-P. Bezemer, Y. Zou, and A. E. Hassan, "An empirical study of game reviews on the Steam platform," *Empir Softw Eng*, vol. 24, no. 1, pp. 170–207, Feb. 2019. https://doi.org/10.1007/s10664-018-9627-4
[4] I. J. Livingston, L. E. Nacke, and R. L. Mandryk, "The impact of negative game reviews and user comments on player experience," in *ACM SIGGRAPH 2011 Game Papers*, New York, NY, USA: ACM, Aug. 2011, pp. 1–5. https://doi.org/10.1145/2037692.2037697
[5] M. Washburn, P. Sathiyanarayanan, M. Nagappan, T. Zimmermann, and C. Bird, "What went right and what went wrong," in *Proceedings of the 38th International Conference on Software Engineering Companion*, New York, NY, USA: ACM, May 2016, pp. 280–289. https://doi.org/10.1145/2889160.2889253
[6] Valve, "Steam Chart - Apex Legends.".
[7] L. Levy and J. Novak, "Planning Your Strategy: Bug Categories, Tools & Documentation," in *Game Development Essentials: Game QA & Testing*, New York: Delmar, Cengage Learning, 2010, p. 77.
[8] A. Lamkanfi, S. Demeyer, E. Giger, and B. Goethals, "Predicting the severity of a reported bug," in *2010 7th IEEE Working Conference on Mining Software Repositories (MSR 2010)*, IEEE, May 2010, pp. 1–10. https://doi.org/10.1109/MSR.2010.5463284
[9] W. Maalej and H. Nabil, "Bug report, feature request, or simply praise? On automatically classifying app reviews," in *2015 IEEE 23rd International Requirements Engineering Conference (RE)*, IEEE, Aug. 2015, pp. 116–125. https://doi.org/10.1109/RE.2015.7320414
[10] I. M. Mika Parwita and D. Siahaan, "Classification of Mobile Application Reviews using Word Embedding and Convolutional Neural Network," *Lontar Komputer : Jurnal Ilmiah Teknologi Informasi*, p. 1, May 2019. https://doi.org/10.24843/LKJITI.2019.v10.i01.p01
[11] K. Phetrungnapha and T. Senivongse, "Classification of Mobile Application User Reviews for Generating Tickets on Issue Tracking System," in *2019 12th International Conference on Information & Communication Technology and System (ICTS)*, IEEE, Jul. 2019, pp. 229–234. https://doi.org/10.1109/ICTS.2019.8850962
[12] H. Zhu, E. Chen, H. Xiong, H. Cao, and J. Tian, "Mobile App Classification with Enriched Contextual Information," *IEEE Trans Mob Comput*, vol. 13, no. 7, pp. 1550–1563, Jul. 2014. https://doi.org/10.1109/TMC.2013.113
[13] A. F. Hidayatullah and M. R. Ma'arif, "Pre-processing Tasks in Indonesian Twitter Messages," *J Phys Conf Ser*, vol. 801, p. 012072, Jan. 2017. https://doi.org/10.1088/1742-6596/801/1/012072
[14] L. Hickman, S. Thapa, L. Tay, M. Cao, and P. Srinivasan, "Text Preprocessing for Text Mining in Organizational Research: Review and Recommendations," *Organ Res Methods*, vol. 25, no. 1, pp. 114–146, Jan. 2022. https://doi.org/10.1177/1094428120971683
[15] T. Kolajo, O. Daramola, A. Adebiyi, and A. Seth, "A framework for pre-processing of social media feeds based on integrated local knowledge base," *Inf Process Manag*, vol. 57, no. 6, p. 102348, Nov. 2020. https://doi.org/10.1016/j.ipm.2020.102348
[16] M. Işık and H. Dag, "The impact of text preprocessing on the prediction of review ratings," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 28, no. 3, pp. 1405–1421, May 2020. https://doi.org/10.3906/elk-1907-46
[17] D. Sarkar, *Text Analytics with Python*. Berkeley, CA: Apress, 2016. doi: 10.1007/978-1-4842-2388-8.
[18] M. Xu, L. He, and X. Lin, "A Refined TF-IDF Algorithm Based on Channel Distribution Information for Web News Feature Extraction," in *2010 Second International Workshop on Education Technology and Computer Science*, IEEE, 2010, pp. 15–19. https://doi.org/10.1109/ETCS.2010.130
[19] M. Alodadi and V. P. Janeja, "Similarity in Patient Support Forums Using TF-IDF and Cosine Similarity Metrics," in *2015 International Conference on Healthcare Informatics*, IEEE, Oct. 2015, pp. 521–522. https://doi.org/10.1109/ICHI.2015.99
[20] M. Alfian, A. R. Barakbah, and I. Winarno, "Indonesian Online News Extraction and Clustering Using Evolving Clustering," *JOIV : International Journal on Informatics Visualization*, vol. 5, no. 3, p. 280, Sep. 2021. http://dx.doi.org/10.30630/joiv.5.3.537
[21] R. Bey, R. Goussault, F. Grolleau, M. Benchoufi, and R. Porcher, "Fold-stratified cross-validation for unbiased and privacy-preserving federated learning," *Journal of the American Medical Informatics Association*, vol. 27, no. 8, pp. 1244–1251, Aug. 2020. https://doi.org/10.1093/jamia/ocaa096
[22] S. Prusty, S. Patnaik, and S. K. Dash, "SKCV: Stratified K-fold cross-validation on ML classifiers for predicting cervical cancer," *Frontiers in Nanotechnology*, vol. 4, Aug. 2022. https://doi.org/10.3389/fnano.2022.972421
[23] G. Alfian *et al.*, "Predicting Breast Cancer from Risk Factors Using SVM and Extra-Trees-Based Feature Selection Method," *Computers*, vol. 11, no. 9, p. 136, Sep. 2022. https://doi.org/10.3390/computers11090136
[24] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, "KNN Model-Based Approach in Classification," in *n The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*, 2003, pp. 986–996. https://doi.org/10.1007/978-3-540-39964-3_62
[25] B. Charbuty and A. Abdulazeez, "Classification Based on Decision Tree Algorithm for Machine Learning," *Journal of Applied Science and Technology Trends*, vol. 2, no. 01, pp. 20–28, Mar. 2021.
[26] W. Zhang and F. Gao, "An Improvement to Naive Bayes for Text Classification," *Procedia Eng*, vol. 15, pp. 2160–2164, 2011. https://doi.org/10.1016/j.proeng.2011.08.404
[27] S. A. Hicks *et al.*, "On evaluation metrics for medical applications of artificial intelligence," *Sci Rep*, vol. 12, no. 1, p. 5979, Apr. 2022. https://doi.org/10.1038/s41598-022-09954-8
[28] I. E. Tiffani, "Optimization of Naïve Bayes Classifier By Implemented Unigram, Bigram, Trigram for Sentiment Analysis of Hotel Review," *Journal of Soft Computing Exploration*, vol. 1, no. 1, Sep. 2020. https://doi.org/10.52465/joscex.v1i1.4
[29] T. Pranckevičius and V. Marcinkevičius, "Comparison of Naive Bayes, Random Forest, Decision Tree, Support Vector Machines, and Logistic Regression Classifiers for Text Reviews Classification," *Baltic Journal of Modern Computing*, vol. 5, no. 2, 2017. http://dx.doi.org/10.22364/bjmc.2017.5.2.05
[30] A. Tye, "FIFA 23 Career Mode Not Working, How to Fix?," Apr. 03, 2023.
[31] A. Mullins, "Apex Legends fans slam 'garbage' anti-cheat after embarrassing hacker clip," Oct. 21, 2021.