Online-Journals.org

iJIM International Journal of Interactive Mobile Technologies

iJIM | eISSN: 1865-7923 | Vol. 19 No. 1 (2025) | OPEN ACCESS

https://doi.org/10.3991/ijim.v19i01.51237

PAPER

Advanced Feature Extraction and Machine Learning Techniques for Classifying Steam Game Feedback

Arar Al Tawil[1](✉), Hanaa Fathi[2], Sharaf Alzoubi[2], Amneh Shaban[1], Laiali H. Almazaydeh[3]

[1]Faculty of Information Technology, Applied Science Private University, Amman, Jordan

[2]College of Computer Science and Informatics, Amman Arab University, Amman, Jordan

[3]College of Engineering, AbuDhabi University, Abu Dhabi, United Arab Emirates

ar_altawil@asu.edu.jo

ABSTRACT

The gaming industry produces vast amounts of user-generated feedback, making it challenging for developers to efficiently analyze and respond to real-time reviews. This study addresses the problem of classifying large-scale unstructured user feedback from Steam reviews. In this paper an approach that integrates traditional machine learning models and deep learning models is proposed. XGBoost is used to manage feature-rich datasets, reducing overfitting. Long-short-term memory (LSTM) and Bi-directional LSTM are used to enhance the accuracy and robustness of classification. Feature extraction techniques, such as sentiment analysis and topic modeling, are employed to enrich the dataset and improve model performance. The experimental results show that the XGBoost model achieved the highest performance with an accuracy of 0.9499 and a ROC-AUC score of 0.6113, demonstrating superior performance in distinguishing between positive and negative feedback. In comparison, deep learning models such as LSTM and Bi-directional LSTM showed lower ROC-AUC scores, indicating less effectiveness in handling the classification task. This approach offers game developers a reliable and scalable solution for classifying user sentiment, leading to better game improvements based on user reviews.

KEYWORDS

machine learning, deep learning, sentiment analysis, feature extraction, topic modeling, natural language processing (NLP)

1 INTRODUCTION

1.1 Background

The game industry is highly developed today [1], and online sales platforms have mainly supplanted traditional offline sales. The user evaluation system of online sales platforms enables players to examine other players' feedback before purchasing. The online commerce game platform has a substantial user base. In 2020, STEAM [2], one of the largest online game retail platforms, reported 120 million monthly active

Al Tawil, A., Fathi, H., Alzoubi, S., Shaban, A., Almazaydeh, L.H. (2025). Advanced Feature Extraction and Machine Learning Techniques for Classifying Steam Game Feedback. *International Journal of Interactive Mobile Technologies (iJIM)*, 19(1), pp. 107–125. https://doi.org/10.3991/ijim.v19i01.51237

Article submitted 2024-07-18. Revision uploaded 2024-10-12. Final acceptance 2024-10-25.

users and 62.6 million daily active users [3]. The game platform receives a substantial number of evaluations from active users. Occasionally, a game may accumulate over 150,000 reviews in a single week and acquire thousands of reviews in a single day [4]. Simultaneously, the complexity of the game software is rising [5], and flaws will inevitably be missed during the company's internal testing. Nevertheless, the online sales platform's hotfix [6] feature enables and motivates game developers to release bug fixes more frequently. This necessitates game manufacturers to read user reviews in real-time and either fix bugs or update the game to prevent a high volume of negative reviews, which may result in a decline in sales [7]. In addition, user evaluations contain valuable information that can assist developers in better comprehending user requirements and complaints during the maintenance and development of the software [8]. Game companies encounter substantial obstacles as a result of the preceding scenario. Analyzing extensive user evaluations in real-time is essential for game companies to identify the flaws that require fixing or the direction of future updates to align with users' needs. Nevertheless, manually perusing a substantial volume of unstructured text, such as user reviews, is a time-consuming and labor-intensive process that imposes substantial expenses on game companies. Therefore, analyzing the requirements of many users can be more efficient if user reviews can be automatically classified based on the types of queries they primarily respond to rather than manually reading each review. Sure, researchers have investigated the classification of user evaluations. Lu and Liang researched the automatic classification of mobile app evaluations [7]. To classify user evaluations of mobile application platforms, they integrated four classification techniques—BoW, TF-IDF, CHI2, and AUR-BoW—with machine learning algorithms Naive Bayes, J48, and Bagging. In their investigation, Rustam et al. implemented a combination of TF-IDF and CHI2 to conclude the sentiment analysis of Shopify user reviews [9]. Nevertheless, conventional classifiers require a significant number of labeled instances for training. The burden on game companies will be exacerbated by the necessity of a significant number of human resources and time to designate instances. Additionally, gaming software garners more user evaluations than other applications [8]. As previously mentioned, the game online sales platforms' review system and hotfix function necessitate that game companies promptly identify reviews that pertain to game defects among the substantial volume of user reviews in contrast to conventional software to prevent user loss [6]. Simultaneously, no research has been conducted to the best of our knowledge, specifically concentrating on categorizing game evaluations to assist the game development industry. The objective is to present a method that enables game companies to conveniently classify game evaluations at a low cost, enabling them to update games more efficiently based on user feedback in light of the factors above.

1.2 Proposed approach

This paper suggests a novel method for efficiently classifying Steam game evaluations by incorporating a variety of advanced machine learning and deep learning techniques. This method is designed to enhance game developers' comprehension of user feedback by integrating traditional machine learning models, recurrent neural networks, and word embeddings, thereby capitalizing on the advantages of each method. Our initial step involves preprocessing the review data. This includes converting all text to lowercase, removing special characters, and cleansing and normalizing the text. A key component of this phase is the use of the VADER sentiment analysis tool. It extracts crucial features such as the review duration, word

count, sentiment scores, and aspect-based counts of positive and negative words. This ensures that the data is standardized and enriched with valuable features. To uncover latent topics within the evaluations, an unsupervised learning approach is employed. Specifically, Latent Dirichlet Allocation (LDA) for topic modeling is implemented. An LDA model is trained to generate topic distributions for each review, which are then used as additional features for classification. This method allows us to identify the underlying themes and topics in the user feedback without the need for labeled data. Additionally, Word2Vec is used to generate word embeddings that encapsulate the semantic relationships between words in the reviews. The dense vector representations for words are obtained by training the Word2Vec model on the tokenized review text. Subsequently, these representations are averaged to create fixed-length feature vectors for each review. These embeddings enhance our feature set and provide a layer of semantic comprehension. A comprehensive feature matrix is constructed by combining the extracted features, which include the review duration, word count, sentiment scores, aspect-based counts, topic distributions, and word embeddings. This matrix is normalized to guarantee that all features contribute equitably during the model training procedure. For evaluation of the efficacy of various classifiers on the classification task, a baseline performance is established. Conventional machine learning models, including support vector machines (SVM), random forest, and XGBoost, are trained on the combined feature set. Furthermore, deep learning models are investigated, such as long-short-term memory (LSTM) and Bi-directional LSTM, that are trained on tokenized and padded sequences of review text to capitalize on the temporal dependencies within the sequences. Additionally, a simple RNN and an artificial neural network (ANN) are implemented to facilitate additional comparisons. Each model is assessed using accuracy, precision, recall, F1-score, and ROC-AUC. Visualizing model performance underscores the effectiveness of our approach through the use of confusion matrices. Our proposed methodology guarantees the interpretability of the classification models and improves their accuracy and robustness. By integrating traditional and neural network models with rich feature extraction and topic modeling, this paper offers a potent instrument for game developers to analyze user feedback more effectively. This method enables the models to be easily adapted and reused in various application scenarios, rendering it a versatile solution for analyzing user feedback forms.

1.3 Contributions

This paper proposes and validates an approach that can classify Steam game reviews to help game developers better analyze user feedback by integrating advanced machine learning and deep learning techniques. This paper extensive feature extraction from review texts, such as sentiment scores, topic distributions using LDA, and semantic embedding with Word2Vec, enhancing the richness and diversity of the feature set.

It explores the combination of traditional machine learning models (XGBoost, random forest, SVM) and deep learning models (LSTM, Bi-directional LSTM, ANN, simple RNN) to evaluate and compare their performance on the classification task, highlighting the benefits of each approach.

According to our evaluation, game companies can reuse our approach with minimal labeled instances, low labor costs, and limited hardware resources, allowing for easy adaptation to different application scenarios. The proposed methodology achieves significant improvements in classification accuracy, providing a robust tool for game developers to analyze user feedback effectively.

2 RELATED WORK

Analysis and classification of user evaluations have been extensively investigated, as evidenced by the following works. According to Rustam et al. [9], app stores typically allow users to provide ratings and evaluations of the applications available in the market. The developers can utilize this information to address issues and devise additional strategies for future updates. Additionally, they believe that to capitalize on this data, the user evaluations must be categorized as positive or negative. To achieve this, they employed a variety of classifiers, including BoW, TF-IDF, and CHI2, to categorize reviews of Shopify applications as either pleased or unhappy.

Guzman et al. [10] proposed a taxonomy for organizing app evaluations into categories pertinent to software evolution. Their experiment, which integrated various machine learning methodologies, yielded promising results. The potential impact of their research on software development is significant, as it could enable developers to prioritize their duties and analyze user reviews more effectively. They believe that their methodology can be expanded to automatically rank the categorized reviews by considering the ratings and sentiments contained within the reviews, thereby prioritizing evaluations with a more negative sentiment or rating. Furthermore, the processing effort could be further reduced by the implementation of mechanisms that summarize and visualize the content of the classified evaluations.

According to Lu et al. [7], the most prominent app distribution platforms have an extensive collection of user evaluations. Research indicates that user evaluations contain a wealth of valuable information that can assist developers in enhancing their applications. Consequently, their objective is to extract and evaluate the non-functional requirements concealed in user evaluations on the app distribution store and characterize a collection of quality attributes desired for an application. The BoW, TF-IDF, CHI2, and classifiers such as Naive Bayes, Bagging, and J48 were combined to classify user reviews into four categories of NFRs. Their findings indicate that the optimal outcome is achieved by combining augmented BoW with Bagging, resulting in a precision of 71.4%.

According to Carreno et al. [11], the enhancement of software quality is heavily reliant on user feedback. They investigate the extensive collection of user feedback available for third-party mobile applications as a method of extracting new or modified requirements for future versions. This study underscores the value of user feedback in the software development process, making the audience feel the importance of their work in enhancing user experience. They believe that the volume of user review data and the time commitment required to extricate new or modified requirements are potential issues. Consequently, they suggested a novel method for automating topic extraction to alleviate some of the process. They extricate the primary topics and a few sentences representing them by processing user evaluations. They stated that this information could help requirements engineers revise the requirements for the upcoming releases. Additionally, their findings indicate that the automatically extracted topics were consistent with those manually extracted.

According to Chen et al. [12], the app market is expanding at an increasing rate. App developers invest significant time and effort in gathering and utilizing user feedback to enhance user contentment. However, they need more effective user review analytics tools. Thus, they suggested AR-Miner, a mobile app review mining framework enabling developers to derive the most “informative” information from raw user reviews in app marketplaces with minimal human effort.

This study [13] examines the application of machine learning techniques to classify clinical text extracted from electronic medical records (EMRs). The authors describe

the challenges encountered during clinical text classification, such as semantic ambiguity, misspellings, irrelevant information, and abbreviations. The proposed methodology comprises text preparation, word representation, feature reduction, and classification. The investigation implements four machine learning methodologies: 1) logistic regression, 2) SVM, 3) naive Bayes, and 4) K-nearest neighbors. These algorithms are incorporated with various word representation models, such as TF-IDF, Word2Vec, and sack of words. The experimental results indicate that the k-nearest neighbor classifier, which utilizes Word2vec representation, achieved the highest accuracy (92%) in accurately classifying medical specialties using transcribed text. The results underscore the importance of high-quality word representation models, such as Word2vec, in enhancing classification performance by preserving contextual and linguistic information. Subsequent research will focus on verifying the proposed methodology using more extensive datasets and investigating deep learning methods to improve performance.

Using a combination of transfer learning (BERT), unsupervised learning, and active learning, Zhang Yejian and Shingo Takada (2023) [14] proposed a machine learning-based method for classifying game user reviews in user review classification. Classification accuracy of 88.8% was achieved with only 100 labeled training instances, thereby addressing the challenge of managing the immense volume of reviews generated on platforms such as STEAM. The method is a cost-effective solution for game developers, as it can be adapted to various game types with minimal additional labeling and substantially reduces the manual effort required for labeling. Their findings illustrate the method's superiority over conventional classifiers, which typically necessitate a greater quantity of labeled data and achieve a maximum accuracy of 62.3%. The potential of this approach to efficiently manage large-scale unstructured text data in the gaming industry is underscored by its extensive use of natural language processing (NLP) techniques and active learning.

2.1 Differences between our proposed approach and related work

The following is a summary of the distinctions between our proposed approach and the research above on the classification of user reviews:

1. Our study differs from previous studies that primarily employ conventional text representation techniques (e.g., BoW, TF-IDF, and CHI2) by incorporating sophisticated feature extraction methods, such as sentiment analysis with VADER, topic modeling with LDA, and word embeddings with Word2Vec. This comprehensive feature set improves the richness and profundity of the information utilized for classification.
2. Our methodology is distinguished by its integration of conventional machine learning models (e.g., XGBoost, Random Forest, and SVM) with deep learning models (e.g., LSTM, Bi-directional LSTM, ANN, and Simple RNN). This hybrid methodology capitalizes on the advantages of both model types, offering a versatile and durable solution for user review classification.
3. Our study is the first to investigate the classification of game reviews from online game sales platforms to the best of our knowledge. This market provides game developers with substantial opportunities to enhance their comprehension and utilization of user feedback, as it boasts an extensive user base and reviews.
4. Our study is designed to significantly reduce the costs of acquiring labeled training data by classifying user reviews with a limited number of labeled instances.

The accessibility and practicality of this approach are further enhanced by its optimization for minimal labor costs and hardware resources, which are suitable for game companies with variable resource levels. This could revolutionize the way game developers approach user feedback. Our primary objective is to organize user evaluations.

5. Our methodology is effective and highly adaptable to a variety of application scenarios with minimal modifications. Game companies can ensure broad applicability and simplicity of implementation by adapting our approach to their specific requirements with small amounts of labeled data and limited resources.

3 PROPOSED METHOD

In this paper, a new approach is introduced to classifying Steam game evaluations that will assist game developers in conducting a more thorough analysis of user feedback. a combination of deep learning and sophisticated machine learning techniques to extract and use a comprehensive set of features from user reviews is presented. Our approach endeavors to expand the depth of the information utilized for classification by incorporating sentiment analysis, topic modeling, and word embeddings. Furthermore, both conventional machine learning models and deep learning architectures are employed to develop a classification system that is both adaptable and robust, capable of effectively managing the diverse and extensive nature of game evaluations.

The overall architecture of our proposed method is illustrated in Figure 1. It outlines the key stages of our approach, including data preprocessing, feature extraction, and the integration of various classification models.

3.1 Dataset

The dataset utilized for this methodology is the Steam Reviews dataset, which is accessible on Kaggle. Steam is the most popular platform for playing games on personal computers. Steam has collected numerous evaluations for its titles over many years on the internet. These reviews provide an exceptional opportunity to analyze the satisfaction and dissatisfaction factors associated with games and genres and sentiment over time. This dataset comprises more than 6.4 million publicly accessible English reviews from the Steam Reviews section of the Steam store, which Valve operates. The review text, the ID of the game to which it pertains, the sentiment of the review (positive or negative), and the number of users who found the review helpful are all included in the description of each review (see Table 1) [15].

Table 1. Key features of the steam reviews dataset

Feature Name	Description
app_id	Unique identifier for each game (Game ID).
app_name	Name of the game (Game Name).
review_text	The actual text content of the review written by the user.
review_score	Review sentiment: whether the review recommends the game or not.
review_votes	Number of helpful votes the review received from other users.

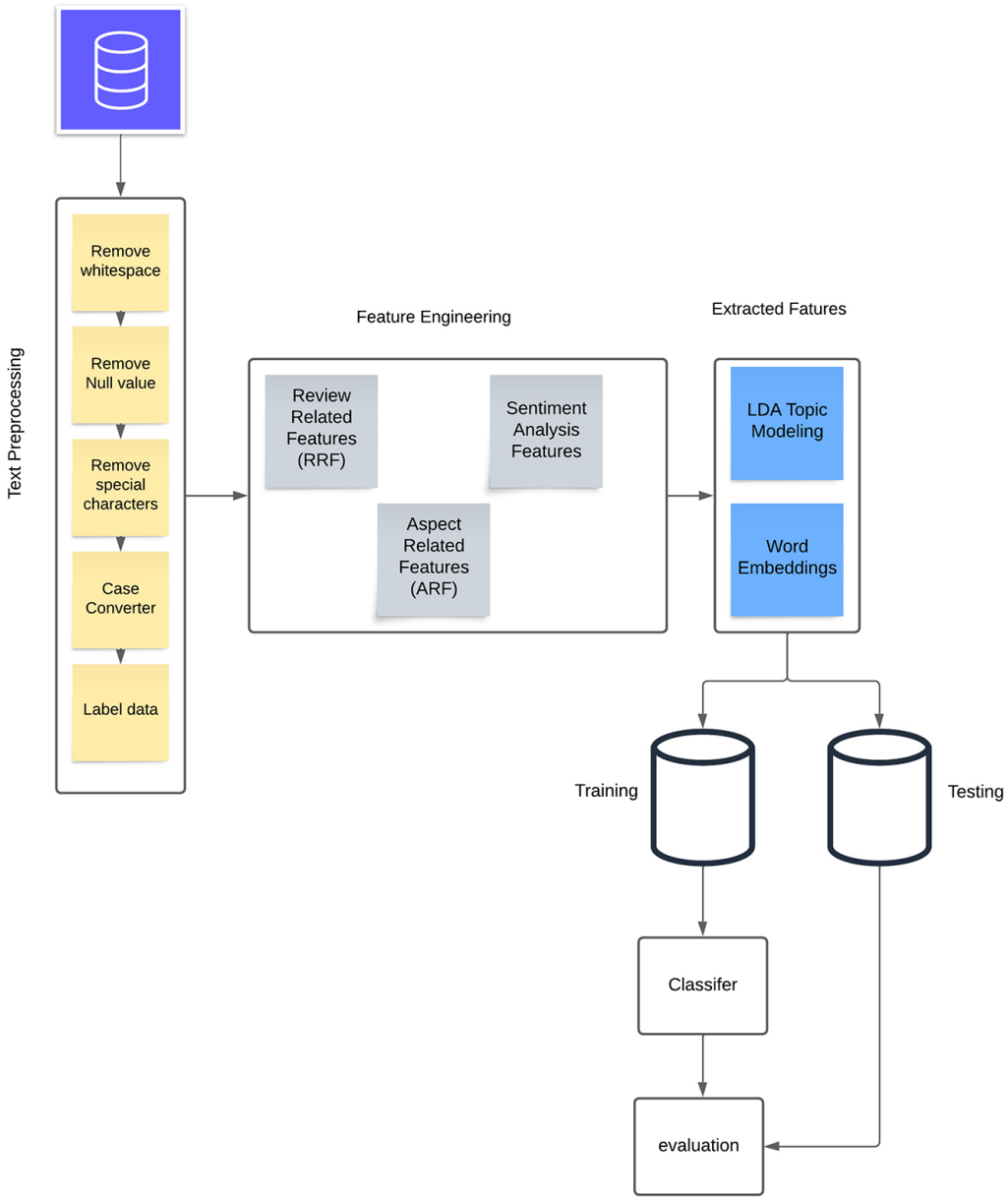

Fig. 1. Overview of the proposed method for classifying Steam game reviews

3.2 Data preprocessing steps

1. **Remove whitespace:** Whitespace characters, including tabs, spaces, and newline characters, do not convey significant information. Consistency and standardization of the data are achieved by eliminating superfluous whitespace from the review text. Whitespace is frequently encountered at the commencement or conclusion of the text and in the form of multiple spaces between words. Removing these can enhance the efficiency of text processing and prevent potential issues in subsequent analyses [16].
2. **Remove null values:** Null values in the dataset may cause errors during model training and analysis. By eliminating entries with absent data, we guarantee that the dataset is comprehensive and that each review contains the requisite information for analysis. Data entry errors or incomplete data acquisition are among

the numerous causes of missing values. These lacking values can result in inaccurate results or errors, as they can cause issues in algorithms that anticipate complete data [17].

3. **Remove special characters:** Special characters, including punctuation marks, numerals, and symbols, do not typically contribute to sentiment analysis. Removing these characters simplifies the text and emphasizes the significance of the words. The text data may be contaminated by special characters, which can introduce noise. By eliminating them, the text's complexity is reduced, and the focus is on the content terms beneficial for sentiment analysis [18].
4. **Convert text to lowercase:** It is crucial to convert the review scores into binary labels in the context of sentiment analysis. In this dataset, review scores are assigned values of 0 (negative sentiment) and 1 (positive sentiment). This conversion simplifies the classification task. The process of labeling data entails the conversion of raw scores or text sentiments into a format that is appropriate for machine learning models. In order to facilitate classification tasks, binary labels are frequently implemented [19].
5. **Label data:** The conversion of review scores into binary identifiers is crucial in the context of sentiment analysis. 0 (negative sentiment) and 1 (positive sentiment) are the mapped values for review scores in this dataset. This conversion facilitates the classification procedure. Data labeling entails the conversion of raw scores or text sentiments into a format that is appropriate for machine learning models. Classification tasks frequently employ binary identifiers for their simplicity [20].

3.3 Feature engineering

Feature engineering is an essential component of the data preparation process for deep learning and machine learning models. It entails converting unprocessed data into meaningful features that can enhance the model's performance. Several feature engineering techniques were employed in this approach to improve the classification of Steam reviews.

1. **Review-related features:** Review-related features (RRF) capture basic characteristics of the reviews. The review length measures the number of characters in the review text, providing a basic indicator of the review's verbosity. The word count counts the number of words in the review text, offering another perspective on the review's length and detail. These features are simple yet informative indicators of the review content's extent [21].
2. **Aspect-related features:** Specific positive and negative words within the review text are the source of aspect-related features (ARF). The positive word count is a metric that quantifies the number of positive words in the review text, which may suggest a positive review. The negative word count is a metric that quantifies the number of negative words in the review text, which may suggest a negative review. These features aid in comprehending the factors that influence the positive or negative sentiment conveyed in the reviews [21].
3. **Sentiment analysis features:** Sentiment analysis features provide a measure of the overall sentiment expressed in the review text. The VADER (Valence Aware Dictionary and sEntiment Reasoner) sentiment analysis tool is used to compute sentiment scores. The sentiment score is a compound score provided by VADER, indicating the overall sentiment of the review text on a scale from –1 (most negative) to 1 (most positive). VADER is particularly effective for analyzing

the sentiment of social media texts and reviews due to its ability to handle various forms of text expressions [22].

3.4 Feature extraction techniques

Feature extraction is the process of converting unprocessed data into a collection of features that can be employed to enhance the performance of machine learning models. It entails selecting, modifying, and constructing variables from the original data, thereby simplifying the interpretation and analysis of models. The primary feature extraction techniques employed in this methodology are as follows:

1. **Topic Modeling:** Topic modeling identifies the fundamental topics in the review text, thereby delivering a more comprehensive contextual representation. The topic distribution feature denotes the topics LDA identified for each review. LDA is a widely used topic modeling technique that is beneficial for comprehending large datasets of text reviews by revealing the concealed thematic structure in a collection of documents [23].
2. **Word Embeddings:** Word embeddings capture the semantic relationships between words by mapping them to high-dimensional vectors. Word2Vec is employed to generate the word embedding features, which generate word embeddings that represent each word in the review text according to its context. Word2Vec preserves semantic similarities and relationships by embedding words into a continuous vector space, thereby capturing the nuances of language [24].

3.5 Feature fusion

The process of feature fusion entails the integration of all engineered features into a single feature matrix. Word embeddings, sentiment score, positive and negative word counts, topic distributions, and review length comprise the combined features matrix. The feature matrix that has been combined is subsequently standardized. Through the integration of a variety of features, the model is able to more effectively classify and comprehend the data, resulting in a more comprehensive representation of the review text.

3.6 Classifiers

This investigation employs a diverse array of deep learning and machine learning classifiers to evaluate their effectiveness in classifying skin cancer lesions. The classifiers implemented are SVM, random forest, XGBoost, an enhanced ANN, and an enhanced LSTM network. Each classifier contributes to an exhaustive model performance evaluation by contributing unique strengths to the classification task.

1. **Support vector machine:** The SVM equation involves the parameters C and σ in the context of the radial basis function (RBF) kernel [25]:

$$k(x_i, x_j) = \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\sigma^2}\right) \quad (1)$$

Here:

- C is the regularization parameter in SVM that controls the trade-off between achieving a low training error and a low testing error.
- σ (gamma in some contexts) is a parameter of the RBF kernel that controls the width of the Gaussian distribution.

The second step is the process of predicting the regression of the SVM model for sampled data. The PSO can be represented by

$$V_i(t+1) = wV_i(t) + c_1 r_1 (P_i(t) - X_i(t)) + c_2 r_2 (G(t) - X_i(t)) \quad (2)$$

Where:

- w is the inertia weight
- c_1 and c_2 are acceleration coefficients
- r_1 and r_2 are random values between 0 and 1
- $G(t)$ is the best-known position among all particles in the swarm

The inertia weight (ω\omegaω) is a key parameter in controlling the trade-off between exploration and exploitation. Adjusting c1c_1c1 and c2c_2c2 influences the impact of personal and global information on particle movement.

2. **Random forest classifier:** Using random subsets of the training data and features, the random forest classifier constructs multiple decision trees and subsequently averages their predictions. Compared to individual decision trees, this ensemble procedure enhances accuracy and minimizes overfitting. A random subset of features is evaluated at each division in the tree, and each tree in the forest is constructed independently using a bootstrap sample of the data. The final prediction is determined by combining the outputs of all trees in the forest, which is typically accomplished through majority voting for classification tasks or averaging for regression tasks. For instance, a random forest model was successful in predicting the outcomes of breast cancer patients who had weak responses to neoadjuvant chemotherapy. This model, which was constructed using a variety of explanatory variables, outperformed logistic regression models in critical performance metrics, including the area under the curve (AUC) [26]. An additional application of environmental science was the prediction of erosive rain events. The random forest model's efficacy over traditional non-machine learning methods was demonstrated by its high accuracy in distinguishing between erosive and non-erosive events while analyzing extensive datasets on rainfall characteristics [27]. Furthermore, random forest classifiers have been implemented in advanced manufacturing to identify anomalies in laser-powder bed fusion processes through optical monitoring. Its high precision and recall demonstrated the model's utility in quality control [28].
3. **XGBoost:** EXtreme Gradient Boosting [29], is a cutting-edge machine learning algorithm renowned for its exceptional performance and efficacy in classification and regression tasks. It functions by sequentially constructing an ensemble of decision trees, with each subsequent tree designed to rectify the errors of the preceding one. This iterative boosting procedure enhances the model's robustness and accuracy against overfitting. XGBoost is highly regarded for its capacity to manage large-scale datasets effectively and its scalability and computational speed. It is a versatile instrument in various machine learning applications, including features such as regularization to prevent overfitting, support for parallel processing, and the ability to manage absent data.
4. **A Bidirectional Long Short-Term Memory (Bi-LSTM):** A bidirectional Bi-LSTM classifier is a sophisticated recurrent neural network (RNN) that improves the

functionality of conventional LSTMs by processing data in both forward and reverse directions. This bidirectional approach allows the model to incorporate patterns and dependencies from both past and future contexts, rendering it particularly effective for sequence prediction tasks such as time series analysis and NLP. The Bi-LSTM model includes a cell state and a variety of gates (input, forget, and output) in each LSTM unit to regulate information flow, thereby addressing long-term dependencies and mitigating the vanishing gradient problem. This dual-layered structure enables Bi-LSTMs to outperform unidirectional models by offering a thorough comprehension of the input sequence [30].

5. **Artificial neural networks:** Artificial Neural Networks (ANNs) are computational models inspired by the structure and functionality of the human brain. They consist of layers of interconnected nodes, or "neurons," that process and transmit information. Each connection between neurons has an associated weight that is adjusted during the learning process. ANNs are capable of learning from data and are used for various tasks such as classification, regression, and pattern recognition. The basic structure of an ANN includes an input layer, one or more hidden layers, and an output layer. The neurons in the input layer receive input data and pass it to the neurons in the hidden layers. These hidden layers apply transformations using activation functions to capture complex patterns in the data. The final output layer produces the prediction.

   The implemented ANN model is designed for binary classification tasks and includes the following key components:
   - Input Layer: This layer takes the input data with a specified shape.
   - First Hidden Layer: It consists of 256 neurons and uses the ReLU (Rectified Linear Unit) activation function to introduce non-linearity into the model. A dropout layer is added after this to prevent overfitting by randomly setting a fraction of the input units to zero during training.
   - Second Hidden Layer: Similar to the first hidden layer, this layer has 128 neurons with a ReLU activation function, followed by another dropout layer for regularization.
   - Output Layer: The final layer contains a single neuron with a sigmoid activation function, making it suitable for binary classification tasks. This layer outputs a probability value indicating the likelihood of the input belonging to the positive class.
   - Compilation: The model is compiled using the binary cross-entropy loss function, which is appropriate for binary classification problems. The Adam optimizer is used for training, which adapts the learning rate during the training process to improve performance. Accuracy is set as the evaluation metric to monitor the model's performance.

   This ANN model leverages dropout layers to mitigate the risk of overfitting and uses ReLU activation functions to allow the network to learn complex patterns. The combination of these techniques ensures that the model can generalize well to unseen data while maintaining high accuracy on the training data [31].

6. **Long short-term memory network:** Long short-term memory networks are a form of RNN intended to address conventional RNNs' constraints, precisely the issue of long-term dependency and the vanishing gradient. LSTMs are well-suited for tasks that necessitate comprehending context over extended periods, such as NLP, time series forecasting, and speech recognition, due to their capacity to learn and recall over lengthy data sequences.

   The memory cell, which regulates the passage of information and maintains its state over time, is the fundamental innovation of LSTM networks. It comprises

three gates: input, forget, and output. By regulating the addition and removal of information to the cell state, the network is able to preserve long-term dependencies and learn relevant patterns from the data.

The LSTM model implemented here is designed for binary classification tasks and includes the following components:

- Embedding layer: This layer converts the input data, which consists of word indices, into dense vectors of fixed size. This helps in capturing the semantic relationships between words.
- Long-short-term memory layer: The core of the model, this layer contains 128 LSTM units that process the input sequences. The LSTM units are responsible for learning temporal dependencies and capturing important features from the sequential data.
- Dropout layer: Added to prevent overfitting, this layer randomly sets a fraction of the input units to zero at each update during training.
- Dense layer: The final layer consists of a single neuron with a sigmoid activation function, which outputs a probability score for binary classification.
- Compilation: The model is compiled using the binary cross-entropy loss function, which is suitable for binary classification problems. The Adam optimizer, known for its efficiency and adaptive learning rate, is used to train the model. Accuracy is set as the evaluation metric to monitor the model's performance.

This LSTM model leverages the capability of LSTM units to maintain long-term dependencies and incorporates dropout to enhance generalization. The combination of these features ensures the model effectively learns from sequential data and achieves high performance in binary classification tasks [32].

7. **Simple recurrent neural network:** A simple RNN is a specific form of neural network specifically designed to analyze sequential data and time series. Simple RNNs, in contrast to conventional feedforward neural networks, possess connections that form directed cycles, which allows them to preserve a form of memory throughout the sequence of inputs. The network can capture temporal dependencies and patterns within the data due to this architecture, rendering it appropriate for tasks such as language modeling, speech recognition, and time series prediction.

   The implemented simple RNN model is designed for binary classification tasks and includes the following components:

   - Embedding Layer: This layer transforms the input data, consisting of word indices, into dense vectors of fixed size. This helps capture semantic relationships between words and reduces the dimensionality of the input space.
   - Simple RNN Layer: The core layer of the model, it consists of 128 simple RNN units. These units process the input sequences and maintain a hidden state that captures temporal dependencies across the sequence.
   - Dropout Layer: Added to prevent overfitting, this layer randomly sets a fraction of the input units to zero at each update during training. This helps improve the model's generalization ability.
   - Dense Layer: The final layer contains a single neuron with a sigmoid activation function, which outputs a probability score for binary classification.
   - Compilation: The model is compiled using the binary cross-entropy loss function, which is appropriate for binary classification problems. The Adam optimizer, known for its efficiency and adaptive learning rate, is used to train the model. Accuracy is set as the evaluation metric to monitor the model's performance.

   This simple RNN model leverages the ability of simple RNN units to capture temporal dependencies, with dropout layers enhancing its generalization by mitigating the risk of overfitting. This combination ensures that the model

effectively learns from sequential data and achieves high performance in binary classification tasks [32].

4 EXPERIMENTAL RESULTS

The experiment results and evaluation metrics are shown in this section. The following characteristics of the PC utilized for the experiments are present: x64-based CPU, Intel(R) Core (TM) i5-9750H, 2.60 GHz 2.59 GHz, Windows 10, 8 GB of RAM. 30% of the data was used for testing, while the remaining 70% was used for training and validation. Python programming is being used by the model to perform.

4.1 Evaluation metrics

Four performance metrics were employed to assess the PO, DE, and GWO algorithms: precision, recall, F-score, accuracy, and AUC. Accuracy is a statistical bias metric that quantifies the percentage of a test's success rate. Low accuracy values suggest a discrepancy between the actual and result sets. Table 2 illustrates the confusion matrix for classification, which represents the classification of the potential outcome of recommending an item to a user. Accuracy employs four test measures.

Table 2. Confusion matrix for classification

	Recommended	Not Recommended
Preferred	True Positive (TP)	False Negative (FN)
Not Preferred	False Positive (FP)	True Negative (TN)

Accuracy. Accuracy measures the proportion of correctly classified instances among the total number of instances. It provides insight into the model's overall performance. The formula for accuracy is [23]:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

The F-score. The F-score, a solitary digit, concisely evaluates a system or model's ability to generate precise optimistic predictions and identify every positive instance. The algorithm integrates two fundamental metrics, namely recall (the capacity to identify all positive cases) and precision (the accuracy of optimistic predictions). By achieving an equilibrium between these two variables, the $F1$ score offers a unified metric for evaluating performance. Elevated values on a scale of 0 to 1 indicate superior performance. It serves as a practical instrument for assessing the efficacy of classification systems. Defined by this formula is the F-score [33]:

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2}$$

Precision. Indicates the proportion of positive cases predicted by the model that turned out to be true. It quantifies the precision with which the model generates affirmative predictions. The formula for precision is [34]:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3}$$

Recall. As with sensitivity, recall quantifies the accuracy with which the model detects true positives. It provides the number of accurate optimistic predictions the model makes relative to the total number of positive cases. The formula for recall is [34]:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4}$$

Area under the curve. Area under the curve represents the area under the receiver operating characteristic (ROC) curve, which plots the true positive rate (sensitivity) against the false positive rate (1-specificity). AUC provides a single scalar value to measure the model's overall ability to discriminate between positive and negative classes. AUC values range from 0 to 1, where a higher value indicates better performance. The AUC can be computed using the following formula [35]:

$$\text{AUC} = \int_0^1 TPR(FPR)d(FPR) \tag{5}$$

Area under the curve is a useful metric for evaluating a model's classification performance, especially in scenarios with imbalanced datasets.

4.2 Experimental results

This section presents the performance evaluation of a variety of machine learning models employed for classification tasks. The models evaluated include XGBoost, random forest, SVM, LSTM, ANN, Bi-directional LSTM, and simple RNN. The evaluation metrics used to compare these models are F1-Score, ROC-AUC, Precision, Accuracy, and Recall. The results are summarized in Table 3 and Figure 2.

Table 3. Performance metrics for different models

Model	Accuracy	Precision	Recall	F1-Score
XGBoost	0.9499	0.9546	0.9941	0.9739
Random Forest	0.9463	0.9470	0.9989	0.9722
SVM	0.9423	0.9424	0.9999	0.9703
LSTM	0.9423	0.9423	1.0	0.9703
ANN	0.9441	0.9452	0.9986	0.9712
Bi-directional LSTM	0.9467	0.9499	0.9959	0.9724
Simple RNN	0.9423	0.9423	1.0	0.9703

The XGBoost model achieved the highest accuracy at 0.9499, along with a high precision of 0.9546, recall of 0.9941, and F1-score of 0.9739. The random forest model followed closely with an accuracy of 0.9463, precision of 0.9470, recall of 0.9989, and F1-Score of 0.9722.

The SVM model had an accuracy of 0.9423, precision of 0.9424, recall of 0.9999, and F1-Score of 0.9703. Both the LSTM and simple RNN models achieved the same accuracy and precision of 0.9423, with perfect recall scores of 1.0 and F1-Scores of 0.9703.

The ANN model showed an accuracy of 0.9441, precision of 0.9452, recall of 0.9986, and F1-score of 0.9712. Lastly, the Bi-directional LSTM exhibited strong

performance with an accuracy of 0.9467, precision of 0.9499, recall of 0.9959, and F1-Score of 0.9724, indicating better class distinction capabilities compared to the unidirectional LSTM and simple RNN models.

The performance of various machine learning models was evaluated using the ROC-AUC metric, which indicates the model's ability to distinguish between classes. The models under consideration include XGBoost, random forest, SVM, LSTM, ANN, Bi-directional LSTM, and Simple RNN. The ROC-AUC scores for each model are visualized in Figure 3.

Table 4. ROC-AUC scores for different models

Model	ROC-AUC
XGBoost	0.6113
Random Forest	0.5433
SVM	0.5011
LSTM	0.5
ANN	0.5270
Bi-directional LSTM	0.5695
Simple RNN	0.5

The XGBoost model achieved the highest ROC-AUC score of 0.6113, indicating a strong ability to distinguish between classes. The Random Forest model followed with a ROC-AUC score of 0.5433. The Bi-directional LSTM also performed well with a ROC-AUC score of 0.5695. The ANN model had a moderate ROC-AUC score of 0.5270. The SVM, LSTM, and simple RNN models had lower ROC-AUC scores of 0.5011, 0.5, and 0.5, respectively, suggesting they were less effective in distinguishing between classes.

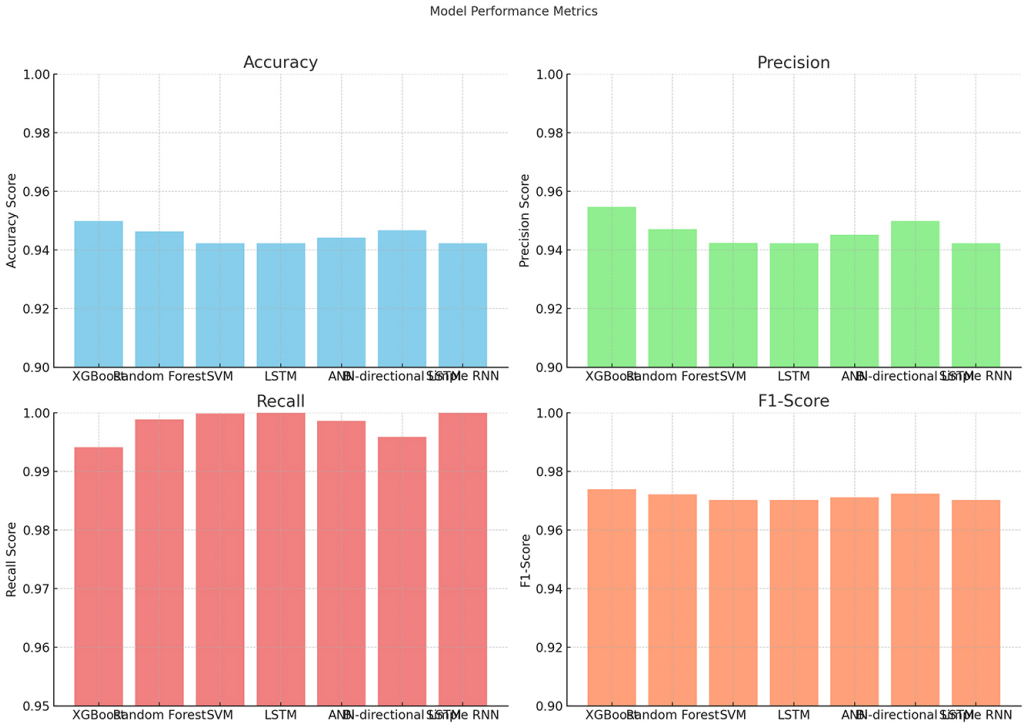

Fig. 2. Comparison of different models based on various metrics

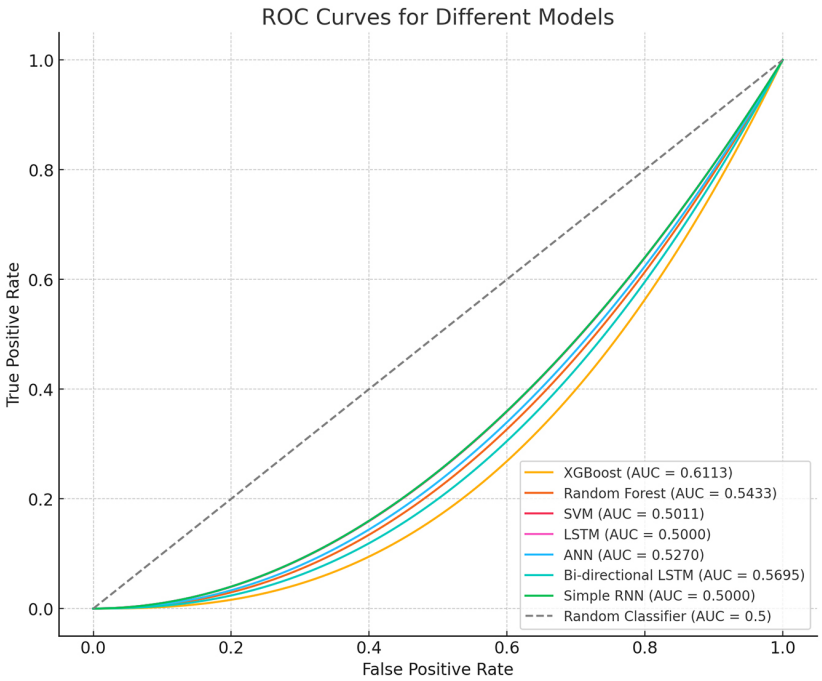

Fig. 3. ROC-AUC comparison across models

5 CONCLUSION AND FUTURE WORKS

This study presented a practical approach for classifying Steam game reviews by integrating traditional machine learning models like XGBoost and random forest with advanced deep learning models such as LSTM and Bi-directional LSTM. The results showed that XGBoost outperformed other models with the highest accuracy (0.9499) and a strong ROC-AUC score (0.6113), highlighting its ability to distinguish between positive and negative feedback. However, handling challenges such as highly unbalanced datasets and noisy data remains critical for future improvements. This study demonstrates the potential of combining traditional machine learning with deep learning techniques for sentiment analysis and large-scale text classification. In future work, oversampling, under sampling, cost-sensitive learning, and data-cleaning methods such as text normalization will enhance model performance. A transformer-based models will be used to compare their performance against traditional models like XGBoost and deep learning models such as LSTM and Bi-directional long-short-term memory.

6 REFERENCES

[1] A. Marchand and T. Hennig-Thurau, "Value creation in the video game industry: Industry economics, consumer benefits, and research opportunities," *Journal of Interactive Marketing*, vol. 27, no. 3, pp. 141–157, 2013. https://doi.org/10.1016/j.intmar.2013.05.001

[2] Valve corporation, "Steam store," 2024. https://store.steampowered.com/

[3] SteamCommunity, "Steamworks development: Steam – 2020 year in review," 2021. Retrieved from https://steamcommunity.com/groups/steamworks/announcements/

[4] Steam, "Cyberpunk2077," 2024. Retrieved from https://store.steampowered.com/app/1091500/

[5] J. Blow, "Game development: Harder than you think: Ten or twenty years ago it was all fun and games. Now it's blood, sweat, and code," *Queue*, vol. 1, no. 10, pp. 28–37, 2004. https://doi.org/10.1145/971564.971590

[6] D. Lin, C.-P, Bezemer, and A. E. Hassan, "Studying the urgent updates of popular games on the steam platform," *Empirical Software Engineering*, vol. 22, pp. 2095–2126, 2017. https://doi.org/10.1007/s10664-016-9480-2

[7] M. Lu and P. Liang, "Automatic classification of non-functional requirements from augmented app user reviews," in *Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering*, 2017, pp. 344–353. https://doi.org/10.1145/3084226.3084241

[8] D. Pagano and W. Maalej, "User feedback in the appstore: An empirical study," in *2013 21st IEEE International Requirements Engineering Conference (RE)*, IEEE, 2013, pp. 125–134. https://doi.org/10.1109/RE.2013.6636712

[9] F. Rustam, A. Mehmood, M. Ahmad, S. Ullah, D. M. Khan, and G. S. Choi, "Classification of shopify app user reviews using novel multi text features," *IEEE Access*, vol. 8, pp. 30234–30244, 2020. https://doi.org/10.1109/ACCESS.2020.2972632

[10] E. Guzman, M. El-Haliby, and B. Bruegge, "Ensemble methods for app review classification: An approach for software evolution (N)," in *2015 30th IEEE/ACM International Conference*, 2015, pp. 771–776. https://doi.org/10.1109/ASE.2015.88

[11] L. V. G. Carreno and K. Winbladh, "Analysis of user comments: An approach for software requirements evolution," in *2013 35th International Conference on Software Engineering (ICSE)*, 2013, pp. 582–591. https://doi.org/10.1109/ICSE.2013.6606604

[12] N. Chen, J. Lin, S. C. H. Hoi, X. Xiao, and B. Zhang, "Ar-miner: Mining informative reviews for developers from mobile app marketplace," in *Proceedings of the 36th International Conference on Software Engineering*, 2014, pp. 767–778. https://doi.org/10.1145/2568225.2568263

[13] L. Almazaydeh, M. Abuhelaleh, A. A. Tawil, and K. Elleithy, "Clinical text classification with word representation features and machine learning algorithms," *International Journal of Online and Biomedical Engineering*, vol. 19, no. 4, pp. 65–76, 2023. https://doi.org/10.3991/ijoe.v19i04.36099

[14] Y. Zhang and S. Takada, "Review classification based on machine learning: Classifying game user reviews," *IEEE Access*, vol. 99, pp. 1–17, 2023.

[15] Kaggle, "Steam reviews," 2021.

[16] G. Angiani, L. Ferrari, T. Fontanini, P. Fornacciari, E. Iotti, and F. Magliani, "A comparison between preprocessing techniques for sentiment analysis in Twitter," in *KDWeb*, 2016. https://www.academia.edu/download/56547431/paper-06.pdf

[17] S. Pradha and M. N. Halgamuge, "Effective text data preprocessing technique for sentiment analysis in social media data," in *2019 11th International Conference on Knowledge and Smart Technology (KST)*, 2019, pp. 1–8. https://doi.org/10.1109/KSE.2019.8919368

[18] A. Krouska, C. Troussas, and M. Virvou, "The effect of preprocessing techniques on Twitter sentiment analysis," in *2016 7th International Conference on Information, Intelligence, Systems Applications (IISA)*, 2016, pp. 1–5. https://doi.org/10.1109/IISA.2016.7785373

[19] E. Haddi, X. Liu, and Y. Shi, "The role of text pre-processing in sentiment analysis," *Procedia Computer Science*, vol. 17, pp. 26–32, 2013. https://doi.org/10.1016/j.procs.2013.05.005

[20] M. S. Shah, M. A. Bhat, M. A. Singh, and M. A. Chavan, "Sentiment analysis," *International Journal of Data Analytics*, 2010. https://www.ijprems.com/uploadedfiles/paper/issue_4_april_2024/33384/final/fin_ijprems1714118825.pdf

[21] G. Kaur and A. Sharma, "A deep learning-based model using hybrid feature extraction approach for consumer sentiment analysis," *Journal of Big Data*, vol. 10, no. 1, 2023. https://doi.org/10.1186/s40537-022-00680-6

[22] C. J. Hutto and E. E. Gilbert, "Vader: A parsimonious rule-based model for sentiment analysis of social media text," in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 8, 2014, no. 1, pp. 216–225. https://doi.org/10.1609/icwsm.v8i1.14550

[23] D. M. Blei, "Probabilistic topic models," *Communications of the ACM*, vol. 55, no. 4, pp. 77–84, 2012. https://doi.org/10.1145/2133806.2133826

[24] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[25] L. Gao, X. Zhang, and F. Wang, "Application of improved ant colony algorithm in SVM parameter optimization selection," *Comput. Eng. Appl.*, vol. 51, no. 13, pp. 139–144, 2015.

[26] Y. Jin, A. Lan, Y. Dai, L. Jiang, and S. Liu, "Development and testing of a random forest-based machine learning model for predicting events among breast cancer patients with a poor response to neoadjuvant chemotherapy," *European Journal of Medical Research*, vol. 28, no. 1, 2023. https://doi.org/10.1186/s40001-023-01361-7

[27] L. Vergni and F. Todisco, "A random forest machine learning approach for the identification and quantification of erosive events," *Water*, vol. 15, no. 12, p. 2225, 2023. https://doi.org/10.3390/w15122225

[28] I. A. Khan, H. Birkhofer, D. Kunz, L. Drzewietzki, and V. Ploshikhin, "A random forest classifier for anomaly detection in laser-powder bed fusion using optical monitoring," *Materials*, vol. 16, no. 19, p. 6470, 2023. https://doi.org/10.3390/ma16196470

[29] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM*, 2016, pp. 785–794. https://doi.org/10.1145/2939672.2939785

[30] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Networks*, vol. 18, nos. 5–6, pp. 602–610, 2005. https://doi.org/10.1016/j.neunet.2005.06.042

[31] F. Chollet, *Deep Learning with Python*. Manning Publications, 2017.

[32] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[33] A. Al Tawil, A. Shaban, and L. Almazaydeh, "A comparative analysis of convolutional neural networks for breast cancer prediction," *Int. J. Electr. Comput. Eng.*, vol. 14, no. 3, pp. 3406–3414, 2024. https://doi.org/10.11591/ijece.v14i3.pp3406-3414

[34] A. Al Tawil, L. Al-Shboul, L. Almazaydeh, and M. Alshinwan, "Fortifying network security: Machine learning-powered intrusion detection systems and classifier performance analysis," *Int. J. Electr. Comput. Eng.*, vol. 14, no. 5, pp. 5894–5905, 2024. https://doi.org/10.11591/ijece.v14i5.pp5894-5905

[35] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, 2006. https://doi.org/10.1016/j.patrec.2005.10.010

7 AUTHORS

Arar Al Tawil earned his BSc in Computer Science from Al-Hussein Bin Talal University, Jordan, in 2018, followed by an MSc from Jordan University in 2021. He currently serves as a lecturer and developer specializing in virtual reality and game design at the Faculty of Information Technology, Applied Science Private University, Amman, Jordan. His professional focus centers on virtual and augmented reality environments, as well as the intersection of machine learning and data analysis. Arar is actively engaged in research exploring new dimensions of technology, particularly in deep learning and NLP. His commitment to advancing these fields reflects his dedication to pushing technological boundaries and contributing to its ongoing evolution. He can be contacted at: ar_altawil@asu.edu.jo.

Dr. Hanaa Fathi received her Ph.D. in Computer Science from Menufia University, Egypt, in 2022, specializing in machine learning and optimization. She is currently an Assistant Professor at the Faculty of Information Technology, Applied Science Private University, Jordan. Hanaa has published more than 10 research papers in international journals and conferences, focusing on feature selection, classification, optimization, machine learning, and web services. She has attended renowned conferences, such as ISACS 2019 and ISCV 2020. She can be contacted at: hanaa_4_ever@yahoo.com.

Dr. Sharaf Alzoubi received his Ph.D. in Computer Science, specializing in mobile computing and software engineering. He is currently an Assistant Professor in the Department of Computer Science at Amman Arab University, Jordan. Sharaf has published extensively in various international journals and conference proceedings, with a focus on optimizing mobile application performance and enhancing user experience through artificial intelligence. His research contributions have earned him recognition for innovative approaches to software development. He teaches courses in Mobile Application Development and Advanced Software Engineering. He can be contacted at: skalzubi@aau.edu.jo.

Amneh Shaban is an esteemed BTEC lecturer specializing in software engineering at the Faculty of Information Technology, Applied Science Private University, Amman, Jordan. She brings extensive experience in leading educational programs, with a strong background as a lead internal verifier (IV), ensuring quality assessments and curriculum development. Amneh holds an MSc in Computer Science from Applied Science Private University, where her research focused on integrating ontology and data mining in smart homes. Passionate about research-driven education, she is dedicated to equipping students with the skills needed to succeed in the field of software engineering. She can be contacted at: a_shaban@asu.edu.jo.

Prof. Laiali H. Almazaydeh received her Ph.D. in Computer Science and Engineering from the University of Bridgeport, USA, in 2013. She is currently a full professor at Abu Dhabi University, UAE. Laiali has published more than 80 research papers in various international journals and conference proceedings. Her research interests include human-computer interaction, and pattern recognition. She has received best paper awards in 3 conferences: ASEE 2012, ASEE 2013, and ICUMT 2016. Recently, she has been awarded two postdoc scholarships from the European Union Commission and the Jordanian-American Fulbright Commission. She can be contacted at: laiali.almazaydeh@adu.ac.ae.