Prediction of user loyalty in mobile applications using deep contextualized word representations

Zeynep Hilal Kilimci

Department of Information Systems Engineering, Kocaeli University, Kocaeli, Turkey

ABSTRACT
Customer loyalty is important for many industries, including banking, telecommunications, gaming, and shopping, in terms of sustainability. In mobile applications, it is observed that the demand rises with the usage of mobile devices such as smartphones. Therefore, it is important to predict when players tend to leave an application. Most of the studies so far focus on churn prediction or customer loyalty in mobile applications by analyzing demographic, economic, and behavioral data about customers. In this work, we introduce sentiment analysis-based customer loyalty prediction in mobile applications using word embeddings, deep learning algorithms, and deep contextualized word representations. To our knowledge, this is the first study to evaluate loyalty of customers analyzing sentiments of users from their comments using deep learning, word embedding, and deep contextualized word representation models. For this purpose, CNNs, RNNs, LSTMs, BERT, MBERT, DistilBERT, RoBERT are used for classification purpose. On the other hand, word embedding models such as Word2Vec, GloVe, and FastText are employed for text representation. To demonstrate the impact of proposed model, comprehensive experiments are performed on seven different datasets. The experiment results show sentiment analysis of users in mobile applications can be a powerful indicator in terms of predicting customer loyalty.

1. Introduction
With the wide use of mobile devices, we have seen a rapid increase in the mobile application industry. Many companies try to enter the mobile application market, where thousands of mobile applications are released every year. The acquisition of new users and adoption of new applications are expensive and require exclusive promotion campaigns to attract users. Like in many markets and businesses, the retention of existing users is more important and cost-effective for a company to keep and increase its revenues. To order to retain current users, the analysis of their behaviour and the prediction of their loyalty is very valuable. Loyalty is especially important in the mobile application industry.
where users have a vast number of alternative applications, a tendency to trial new applications, and easy divergence of user attention to another application. User ratings and reviews provided by application stores contain valuable information for both application developers and potential new users. User reviews of mobile applications frequently contain ratings, suggestions, recommendations, or complaints, which are valuable for application developers to improve user satisfaction and stop and/or decrease the churn rate of existing users.

The notion of customer loyalty is a measurement of a customer’s probability to recur business with a company. Thus, it can be said that it is the outcome of customer satisfaction, experiences of favourable customer, and the overall worth of the services or goods a customer buys from a business. It is known that customers are not readily affected by the pricing or presence of the product or service when a customer is loyal to a certain brand or company. Even customers are eager to repay more on condition that the similar or same quality is presented to them are accustomed. A loyal customer is not effectively looking for different products or companies. They are more eager to advise a brand to their family and friends. They also approach positively the other goods of the brand. Moreover, loyal customers are more insightful if troubles happen, and they have confidence in the brand about fixing problems. It is known that they hold buying from brand/company whenever there is a necessity. Moreover, loyal customers are willing to give feedback on how a company/brand can enhance its services or products. Thus, we consider that there can be a connection between the sentiment analysis of customers from their feedbacks and customer loyalty. On the other hand, marketers measure loyalty by looking at customer behaviour with different metrics such as lifetime value (total amount of money customers spend), churn rate (brand cancellation or disengaged rate of customers), referrals (the number of new customers who register according to recommendations), and net promoter score (intention of customers to mention others about the brand). Unlike these metrics, we propose to analyse same customers/users’ feedbacks that is above a certain number in order to get an idea about their customer satisfaction. In this way, customer loyalty can be detected. Because there is a strong relationship between customer satisfaction and loyalty as mentioned in literature studies (Hallowell, 1996; Rahim et al., 2012). Since customer satisfaction is reflected in comments related to the product, company, or application, we consider that analyzing sentiments on these feedbacks is a significant attempt to measure customer loyalty. On the other hand, the user can be faithful to a particular application based on features that are never mentioned in the comments. In our study, we propose to measure users’ loyalty to the app by interpreting the features mentioned in the comments.

As a natural language processing technique, sentiment analysis allows the identification and categorizations of emotions expressed in a piece of text. Sentiment analysis is widely applied to user reviews and surveys expressed in the text to extract opinions or sentiments of users about a product or service. Application stores such as Google Play, Apple AppStore provide user ratings and reviews about mobile applications. In our previous paper (Kilimci et al., 2020), we presented sentiment analysis-based churn prediction in mobile games using word embedding models and deep learning algorithms. The aim of this paper is to apply sentiment analysis methods for the estimation of user loyalty on data collected from the application store’s ratings and reviews. Datasets are represented by using Word2Vec, Glove, and FastText word embedding models. Three
deep learning algorithms and four deep contextualized word representations are employed for the loyalty prediction. While Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), and Long Short-Term Memory (LSTMs) are used as deep learning algorithms, bidirectional encoder representations from transformers (BERT), Multilingual BERT (MBERT), DistilBERT (DBERT), Robustly Optimized BERT (RoBERTa) are employed as deep contextualized word representations in this work.

This paper is organized as follows: Section 2 makes a review of related work on loyalty prediction of users and deep contextualized word representations. Section 3 presents the methodology and methods used in this study that introduces word embedding models, deep learning techniques, deep contextualized word representations, data collection and proposed model. Section 4 presents experimental results. Section 5 includes a discussion and conclusion.

2. Related work

In this section, loyalty analysis and prediction have been widely studied in several domains including telecom, finance, retail, e-commerce, and banking (Huang et al., 2012; Kumar & Chandrakala, 2016; Santharam & Krishnan, 2018). Most of these researches rely on machine learning and times series forecasting techniques that tries to make a prediction using data samples consisting of tens or hundreds of features. Several researchers have studied loyalty and churn analysis and prediction in mobile games.

In a recent study, Liu et al. (2020) developed loyalty prediction methods at the micro-level (between an application and a specific user) and macro-level (between an application and all its users). At the micro-level, a novel semi-supervised and inductive embedding model is used for loyalty prediction. At the macro-level, they use a method called SimSum based on unbiased estimation. Furthermore, link analysis algorithms PageRank and HITS are also adapted to make loyalty prediction at the macro-level. They evaluated the performance of their methods on a dataset provided by the Samsung Game Launcher platform. This is a large dataset that stores information about tens of thousands of mobile games and hundreds of millions of user-application interactions. In the study (Kristensen & Burelli, 2019), authors present several LSTM based neural network architectures that use both sequential and historical data to predict loyalty prediction by analyzing user behaviour. The dataset contains player logs of a mobile game. In another study (Kim et al., 2017), authors used three traditional machine learning algorithms and two deep learning algorithms by analyzing player log data of three casual games in both the observation period and churn prediction period. Recently, some of the researchers modelled loyalty and churn prediction as a survival analysis problem (Periáñez et al., 2016; Viljanen et al., 2017). Some of the previous research on churn prediction include studies (Hadiji et al., 2014; Tamassia et al., 2016; Xie et al., 2015, 2016).

In another work Huang et al. (2012) concentrate on the loyalty prediction in telecom communication sector by investigating too many features related to analysis of customer behaviour. For this objective, authors employ conventional machine learning models and multi-layer perceptron. They conclude the study that support vector machine and decision tree provide more effective prediction results. In the study (Castro & Tsuzuki, 2015), authors propose to forecast the loyalty of online game players by evaluating a frequency analysis method. They employ a proposed frequency analysis method for feature
representation from login records instead of demographic, economic, and behavioural data about their customers in order to predict players’ loyalty. To measure the success of proposed model, k-nearest neighbours’ algorithm is evaluated. They report that the proposed framework ensures more than 20% profitability. In the study (Milošević et al., 2017) authors present a study-related early churn estimation model in social mobile games. This means that they analyse the loyalty of players one day after registration in free-to-play games. To assess the success of the proposed approach, they utilize machine learning models and compare performances on an online mobile game. They report that the reduction in the churn of players reaches 28%, which stands for an important indicator for a business.

In the study (Liu et al., 2018), authors design a new model for loyalty prediction of mobile games by blending semi-supervised and inductive embedding design. The proposed model is implemented with deep neural networks in addition to modelling a new random walk method. To show the efficiency of a new design, they use real-world data set that is gathered from the Samsung Game Launcher platform. They conclude the study that the proposed approach outperforms state-of-the-art results. In another study (Runge et al., 2014), authors focus on the forecasting loyalty of high-value game players. Four different classification techniques were used for this purpose, in addition to the hidden Markov model. Furthermore, they intend to exhibit the effects of churn of high-value game players on the business side. Experiment results show that the proposed approach tackles the loyalty prediction of significant players. In another work (Kawale et al., 2009), authors investigate the social impact in loyalty estimation for multiplayer online games. A hybrid loyalty estimation design is presented by blending social effect and personal engagement. They declare that the proposed framework significantly boosts the classification success compared to the conventional methods for predicting loyalty or churn in multiplayer online games.

In the study (Moedjiono et al., 2016), authors focus on to predict customer loyalty with the intent of boosting classification accuracy with group customer data into segmentation. For this purpose, decision tree and k-means algorithms are applied to the collected dataset. They report that C4.5 classification algorithm outperforms k-means with 79.33% of accuracy. In the other study (Buckinx et al., 2007), authors concentrate on enhancing the customer database with an estimation of loyalty by observing customer behaviours. For this purpose, two widely used conventional machine learning algorithms are employed, namely random forests and automatic relevance determination neural networks. They conclude the paper that diversity of purchase is the best performing estimator of behavioural loyalty, and that a customer’s expenditure frequency, response to promotion, answer to mailings, and continuity of purchasing all deliver helpful knowledge to deliver an ample estimation of behavioural loyalty of customers. In the study (Wijaya & Girsang, 2016), authors analyse for predicting customer loyalty by employing three data mining techniques, namely decision tree, naive Bayes, nearest neighbour algorithm. They collect real-world data from the national multimedia company in Indonesia, including 2269 records. Experiment results show that the decision tree algorithm outperforms others with 81% of accuracy. It is followed by naive Bayes with 76% of accuracy and nearest neighbour with 55% of accuracy.

In the study (Wassouf et al., 2020), a method is delivered for telecom companies to aim for different-value customers by specific proposals and services. In order to demonstrate the efficiency of the proposed methodology, a dataset with about 127 million records is
provided. As a first step, customers are categorized by implementing a novel approach time–frequency monetary which determines the level of loyalty. Then, segments are assigned. Next, random forest, Decision tree, Gradient-boosted tree, and multiplexer perceptron machine learning algorithms are applied to the segmented dataset. Authors conclude the study that loyalty reasons of customers at each segment are detected and the best proposals are presented to the customers. In another study (Sulistiani & Tjahyanto, 2017), machine learning techniques are evaluated to demonstrate the efficiency of different feature selection methods on customer loyalty. Gain ratio, chi-square, and information gain are appraised as feature selection methods to get the best meaningful features for the dataset of fast-moving consumer goods. Experiment results show that the chi-square method exhibits the highest classification performance with 83.2% of accuracy when the model is classified with a random forest algorithm.

Our study differs from aforementioned literature works as well mainly because we do not employ for analyzing loyalty or churn prediction with common approaches such as analyzing demographic, economic, and behavioural data about their customers. By evaluating sentiment analysis-based mobile application loyalty, we present a different solution compared to the traditional methods that can be a powerful indicator for estimating customer loyalty or churn.

3. Methodology
This section presents a review reproduced from the study (Kilimci et al., 2020) about word embedding methods and deep neural networks. The methodology part is also included sampling techniques used for imbalanced data sets, and deep contextualized word representations used in this study. In this section, the dataset and proposed model are also introduced for loyalty prediction.

3.1. Word embedding models
Word embedding models allow the representation of words as vectors of numerical values. In these models, the words with similar meanings have similar vector representations. A corpus with a large collection of documents is used for learning the vector representations of words. Word embedding models can learn semantic, syntactic similarity, and relationships among words in a given context from a corpus. Although the study of word embedding models started with the studies (Bengio et al., 2003; Collobert & Weston, 2008), these models became popular after Word2Vec model developed by Mikolov and his colleagues at Google. After this study, a large number of new embedding models have appeared in the literature (Bojanowski et al., 2017; Mikolov, Chen, et al., 2013; Mikolov, Sutskever, et al., 2013; Peters et al., 2018). In this study, words are represented by three well-known word embedding models: Word2Vec (Mikolov, Chen, et al., 2013), GloVe (Pennington et al., 2014), and FastText (Joulin et al., 2016).

**Word2Vec.** Word2Vec is the most influential embedding model that can learn word embeddings using a neural network. It is proposed by Le & Mikolov (2014), Mikolov, Chen, et al. (2013), and Mikolov, Sutskever, et al. (2013) at Google. Google released a toolkit that allows training of word embeddings from text and document, and the
usage of pre-trained embeddings. The toolkit enables an efficient implementation of two different word embeddings models: Skip-gram model and continuous bag-of-words (CBOW) model. Skip-gram model tries to predict the nearby words of a given word in a text or document. In CBOW model, the objective is to predict a target word \( w_t \) from the nearby \( n \) words.

**Glove.** GloVe (Pennington et al., 2014) is another popular word embedding model suggested after Word2Vec model. It stands for ‘Global Vectors’. A local content window is passed over training data to learn the semantics of words in the Word2Vec model. Pennington et al. criticized the Word2Vec model that only passes a local content window to obtain semantic relationships among words. It does not exploit the count-based statistical information regarding word co-occurrences. To get a better representation, Pennington et al. proposed the GloVe model that joins the count-based matrix factorization and local content window techniques. GloVe model obtains a global word-word co-occurrence statistic from a corpus by using matrix factorization.

**Fasttext.** FastText is another word embedding model suggested (Joulin et al., 2016; Mikolov et al., 2017). While Word2Vec and Glove utilize whole words as the smallest part, FastText utilizes character \( n \)-grams to represent parts of words as the smallest element. Character \( n \)-grams are sequences of subwords obtained by taking \( n \) consecutive characters from parts of words. Each word is separated into a number of character \( n \)-grams where \( n \) is usually \( 3 \leq n \leq 6 \). For example, when \( n = 3 \), the word ‘model’ is divided into \(< mo, mod, ode, del, and el>\), plus. FastText uses the skip-gram model of the Word2Vec method with a modified loss/cost function. It also applies a negative sampling approach that improves the efficiency of the algorithm by reducing the number of training samples during training.

### 3.2. Deep learning techniques

This section introduces the applied following deep learning methods for loyalty prediction: convolutional neural networks (CNNs), Recurrent Neural Networks (RNNs), and Long Short-Term Memory Networks (LSTMs).

**Convolutional neural networks.** In many images and speech recognition problems, convolutional neural networks (CNNs) have demonstrated a notable performance increase and attracted considerable attention in recent years (Krizhevsky et al., 2017; LeCun et al., 2015; Schmidhuber, 2015; Voulodimos et al., 2018). It was discovered that CNNs are also useful in natural language processing (NLP) tasks (Kim, 2014; Otter et al., 2020; Zhang et al., 2015). CNNs are similar to ordinary multi-layer neural networks with multiple layers. The layers of CNNs are generally composed of input, convolution, pooling, and fully connected layers. Several convolutional layers are interleaved by pooling layers. The convolution layer, being the most important block, applies a convolution filter on its inputs to extract feature maps (features). The convolution operation using multiple filters is used to learn specific features of input data. An extra activation function, RELU (rectified linear unit), is generally employed to feature maps to expand the nonlinear properties of the network. Pooling layers apply pooling operation to reduce the dimensionality of data
coming from the convolution layer. It retains the most useful information after the convolution operation. The pooling operation performs subsampling that decreases the training time provides dimensionality reduction, and prevents over-fitting. After a series of conventional and pooling layers, a flattened input obtained from the last layer is fed into the fully connected layer. The last decision is given by fully connected layers. A regularization method termed ‘dropout’ can be utilized to fully connected layers to decrease overfitting.

**Recurrent neural networks.** RNNs are a class of neural networks that are usually suited for modelling time-series and sequential data. Given an input vector, the RNN output vector does not only influenced by its input but also on the entire history on past input information kept in its hidden states (Elman, 1990; Lipton et al., 2015). RNNs are not like feed-forward networks that do not have loops. The feedback loops in an RNN architecture enable the past decision information to be held in the network. Gradient vanishing and exploding problem are often experienced in RNN architectures. RNNs can lose its ability to learn past data when there are long-term dependencies in sequence data. This inability comes from the gradient descent procedure while training when the gradients are being transmitted back across many layers. Small gradient values shrink exponentially and vanish because of continuous matrix multiplications for the computation of gradient. This is called vanishing gradient problem that makes the model difficult to learn from previous data in a sequence. Continuous matrix multiplications can also explode the gradient values instead of vanishing. The gradient values become larger and go to NaN (not a number) values in the course of training and crash the network to learn. This is described as the exploding gradient problem. A number of methods like gradient clipping or suitable activation functions can be applied to cope with vanishing and exploding problems.

**Long short-term memory networks.** Long Short-Term Memory networks (LSTMs) are variants of RNNs. LSTMs are explicitly designed to overcome the gradient vanishing/exploding problems of RNNs (Graves et al., 2013; Greff et al., 2016; Kent & Salem, 2019). LSTM continues to learn over many time steps by preserving the gradient values go

![Figure 1. A chain of LSTM units.](image-url)
back-propagate through deeper layers. Basically, LSTMs are developed to capture long-term dependencies and relations in the sequence data. The LSTM architectures have the ability to hold the contextual semantics of information and long-term dependencies between data. An LSTM architecture contains a chain of LSTM units or memory cells, as shown in Figure 1. Each LSTM unit has three gates that update and control the cell states. These are input, forget and output gates that control which portions of information to remember, forget and pass to the next step. An LSTM unit gives a decision about what to store and when to permit reads, writes, and deletions via gates while passing or blocking information through an LSTM cell.

3.3. Deep contextualized word representations

Bidirectional encoder representations from transformers (BERT). BERT is known as the frontier of a new language representation, which stands for bidirectional encoder representations from transformers. Distinct from language representation methods, BERT is modelled to pretrain deep bidirectional representations from untagged text by stipulating on both left and right context in all layers. That is, it is trained for the purpose of language understanding on a large text corpus and employed on natural language processing tasks. Moreover, BERT is the first model that internalizes the unsupervised learning approach and deeply bidirectional system. This means that the training procedure with the BERT model is implemented on the raw text data which is the novelty in the NLP task. Pretrained representations can also either be context-free or contextual, and contextual representations can further be unidirectional or bidirectional. Context-free models like Word2Vec or GloVe produce only a single ‘word embedding’ representation for each term in the vocabulary, while contextual models instead produce a representation of each word that is based upon the other terms in the sentence (Devlin et al., 2018).

Multilingual BERT (MBERT). MBERT is an extended version of the BERT model, which is put forward to generalize representations across languages. BERT is released for a single language model pre-trained on the concatenation of monolingual Wikipedia corpora from 104 languages, while M-BERT is trained on the Wikipedia pages of 104 languages with a shared word piece vocabulary. M-BERT enables a very straightforward approach to zero-shot cross-lingual model transfer in place of being trained just on monolingual English data with an English-derived vocabulary. Thus, it facilitates to transfer between languages written in different scripts (Pires et al., 2019).

DistilBERT (DBERT). DistilBERT has the same structure as BERT. While most prior studies analyse the utilization of distillation for constructing task-specific models, knowledge distillation is leveraged during the pretraining stage and demonstrate that it is probable to decrease the size of a BERT technique by 40%, while keeping 97% of its language comprehension skill and being 60% faster. A triple loss combining language modelling, distillation, and cosine-distance losses are leveraged the inductive biases learned by larger methods along pretraining. Thus, it is possible to pretrain smaller, faster, and lighter compared to other models in DBERT (Sanh et al., 2019).

Robustly Optimized BERT (RoBERTa). RoBERTa is proposed as an optimized version of the BERT pretraining model, which contains an assessment of the impacts of
hyperparameter tuning and training set size. RoBERTa is capable to compete or exceed the performance of all of the post-BERT techniques. RoBERTa architecture provides training procedure longer with bigger batches over more data. Next, sentence estimation is removed. Moreover, it is capable to change the masking pattern dynamically performed on the training data. Finally, it facilitates training procedures on longer sequences (Liu et al., 2019).

3.4. Data sampling techniques for imbalanced class distribution

Imbalanced datasets are known as a severe skew in the class distribution in the minority or majority class. This bias in the training dataset can affect the classification performance of machine learning algorithms. This can cause to be neglected of the minority class, completely. This is a challenge as it is generally the minority class on which estimations are most critical. There are many widely used techniques to address the challenge of class imbalance. One of them is randomly resampling the training dataset. The two basic concepts to randomly resampling an imbalanced dataset are to duplicate instances from the minority class, called oversampling and to remove instances from the majority class, called under sampling. In this work, we focus on the random under sampling (RUS) method to delete instances from the majority class and word embedding substitutions (WES) in order to duplicate instances from the minority class. In WES, approach, a number of documents are selected to be duplicated from documents belonging to the minority class. New documents are generated by replacing the words contained in the duplicated documents with synonyms, if any.

3.5. Data collection and proposed model

In this work, we use seven different English datasets that include user comments of related mobile applications. These are Hitman, Soccer Manager, Grand Theft Auto (GTA): San Andreas, City Driving 3D, Amazon, AliExpress, and Ebay. The first four data sets (Hitman, Soccer Manager, GTA San Andreas, City Driving) that are employed in the study (Hallowell, 1996) as Turkish, are translated into English by using Google Translate API. The new data sets (Amazon, AliExpress, Ebay) are collected in English.

Hitman is an action-adventure-focused stealth video game in which players command Agent 47, a genetically boosted assassin, from a third-person perspective, as he travels to international places and performs contracted assassinations of criminal objectives across the world. Soccer Manager is a browser-based multiplayer game that lets managers to buy and sell players and compete head-to-head against other managers from across the world. Grand Theft Auto: San Andreas is an action-adventure-focused game with role-playing and stealth elements. City Driving 3D is a stress buster game that gives the players the feel of driving in a real car in a real city. Ebay is one of the largest online auction shopping sites. AliExpress, which contains many product groups from A to Z, can be used in many countries of the world. Similar to Amazon, AliExpress has a chance for small manufacturers to reach consumers directly.

This study is based on three main stages in order to analyse sentiment analysis-based loyalty prediction of users for each application. These are data collection, data pre-processing, and classification. In data collection step, comments of users are gathered from
Google Play Store pages through Selenium crawler (Bruns et al., 2009). Because of the number of comments downloading limitations of Apple Store, Google Play Store is used at the data collection step. The user comments of the applications are downloaded from html classes and saved into csv file. After collecting user comments with the usage of Python programming language, rating, user name of related player, date, vote, comment information is achieved for each mobile application. In this step, nearly 850,000 comments for Hitman, approximately one million feedbacks for Soccer Manager, almost 605,000 comments for GTA San Andreas, nearly 272,000 feedbacks for City Driving, about 2 million comments for Amazon, approximately 11 million comments for AliExpress, almost 3 million feedbacks for Ebay are totally gathered. In the second step, the text-based raw datasets are cleaned by pre-processing methods because of the dirtiness of the datasets before starting the classification step. In order to remove emojis, punctuation marks, stop words, Zemberek is employed. User comments are rated in the range between 1 and 5. To analyse the sentiment of the comment released by the users, 1, 2, and 3 are evaluated as negative sentiment and the remaining is appraised as positive. After pre-processing, the statistics of the dataset are presented in Table 1. Moreover, Table 1 demonstrates the number of different comments made by the same user three times and above. In order to analyse customer loyalty, the statistics of the data sets located in Table 1 are employed in the experiments. In Table 2, the statistics of comments based on the number of comments of the same users are demonstrated. In all data sets, the most comments are released by the same users three times. The number of comments made four times is also significantly larger. The number of comments made five, six, and seven times by the same user shows a significant decrease. According to the statistics in Table 2 and as mentioned earlier, loyal customers are willing to give feedback on how a company/brand can enhance it services or products.

At the next step, deep learning algorithms, word embedding techniques, and deep contextualized word representations are employed for the purpose of evaluating the sentiment analysis-based loyalty prediction of users. For this intention, convolutional neural networks, recurrent neural networks, long short-term memory networks are used as deep learning models, Word2Vec, GloVe, FastText are utilized as word embedding techniques and BERT, MBERT, DBERT, RoBERTa are employed as deep contextualized word representations, in this study.

Word embedding models present the word vectors while deep learning models perform the classification task. In order to include the impact of word embedding models, the output of each word embedding model is fed into the multi-layer perceptron. Thus, the classification task is also performed for word embedding techniques. Word embedding models are employed for sentiment analysis due to its functionality.

| Dataset          | #Positive | #Negative | Total   |
|------------------|-----------|-----------|---------|
| Hitman           | 17,694    | 3082      | 20,776  |
| Soccer Manager   | 3165      | 2546      | 5711    |
| GTA San Andreas  | 3343      | 1503      | 4846    |
| City Driving     | 7331      | 3380      | 10,711  |
| Amazon           | 49,855    | 7701      | 57,556  |
| AliExpress       | 63,135    | 8950      | 72,085  |
| Ebay             | 59,854    | 9748      | 69,602  |

Table 1. Statistics of the datasets before data sampling.
Because word embedding models provide a word vector in a document which holds the similarity between other words in a document. This means that the usage of word embedding models ensures the syntactic and semantic similarity and correlation among terms in a dedicated context from the corpus. In order to implement models, Keras package is used. We use in the experiments following experiment settings. Epoch number is 25, learning rate is 1e-3, learning lambda rate is 1e-3, batch size is 256, class number is 2 (0 for negative comments and 1 for is positive comments). Furthermore, rectified linear unit (ReLU) is utilized as an activation function. The maximum pool size is adjusted as 2 and GlobalMaxPooling approach is used for pooling operation. The activation function of a fully connected layer is set as sigmoid. To avoid the over-fitting challenge, we apply dropout as 0.2, early stopping-criteria, and weight constraint by adjusting 4.

Moreover, Bert is implemented using two different Python libraries, PyTorch and ktrain. Huggingface transformers are employed because it provides general-purpose architectures such as BERT, GPT-2, RoBERTa, XLM, DistilBert, XLNet, etc. In the Pytorch BERT implementation, the maximum word length is chosen as 200. Batch size is set to 20 and epoch size is adjusted as 5. AdamW is the optimization method used in this work. For MBERT model, bert-base-multilingual-uncased model is used at the implementation stage. Maximum word length is set to 128, batch size is 16, learning coefficient of the model is 1e-4, and epoch size is adjusted as 5. In DBERT, distillbert-base-uncased model is used from the Huggingface library. Maximum word length is arranged as 512, batch size is 16, learning coefficient of the model is 1e-4, and epoch size is 3. In RoBERTa, roberta-base module is obtained from the Huggingface library. Maximum word length is 128, batch size is 8, learning coefficient of the model is 1e-5, and epoch size is set to 5. BERT-related models are conducted with TPU, which is produced personally by Google and whose performance is also superior to the GPU.

After implementing models at different training set percentages, the classification accuracies are enhanced with data sampling techniques in order to eliminate data imbalanced problem. For this purpose, word embeddings substitution (WES) and random under sampling (RUS) techniques are employed for each dataset. The detailed statistics of datasets after data sampling are mentioned in Section 4.

### 4. Experiment results

In this section, the classification performances of deep learning techniques, word embedding models and deep contextualized word representations are evaluated in terms of different training set percentages such as 80, 50, and 30. Training set percentage is expressed as ‘ts’ abbreviation in this part. As a first attempt, we analyse the effect of all
aforementioned models on seven different English mobile application datasets at ts80, as shown in Table 3.

In Table 3, it is observed that FastText exhibits superior classification success with 83.59% of mean accuracy among word embedding models and deep learning algorithms when average classification accuracies are considered at ts80. It is followed by RNN, CNN, Word2Vec, GloVe, LSTM with 80.98%, 80.89%, 77.93%, 75.44%, 74.94% accuracies, respectively. When deep learning techniques are compared, there is no superiority among CNN and RNN methods due to the closeness of classification performances each other while the LSTM model exhibits the poorest performance in order to classify sentiments of users. Thus, the choice of CNN or RNN model for the purpose of categorizing sentiment of players demonstrates a similar impact on aforementioned datasets. When mean classification success of word embedding models is considered among themselves, FastText outperforms Word2Vec and GloVe with approximately 6% and 8% enhancement, respectively. The contribution of each word embedding model to the system can be summarized in the following order: FastText > Word2Vec > GloVe. Moreover, DBERT exhibits the most successful classification performance with 85.34% of accuracy among deep contextualized word representations when mean accuracy is considered. It is followed by RoBERTa with 84.35% of accuracy, BERT with 83.45% of accuracy, and MBERT with 83.16% of accuracy. In addition to deep contextualized word representation methods, DBERT is the highest performing model with an accuracy of 85.34%, considering not only deep contextualized word representation but also deep learning and traditional word embedding models in terms of mean accuracies at ts80.

The classification success of all techniques is represented at ts50 in Table 4. Similar to the results of Table 3, the best classification accuracy is performed by FastText with

| Table 3. The classification accuracies of all techniques without resampling at ts80. |
|---|---|---|---|---|---|---|---|---|
| Models | Hitman | Soccer Manager | GTA San Andreas | City Driving | Amazon | AllExpress | Ebay |
| CNN | 87.89 | 72.17 | 78.52 | 76.00 | 85.32 | 82.21 | 84.15 |
| RNN | 88.41 | 71.78 | 75.80 | 77.24 | 87.57 | 80.66 | 85.42 |
| LSTM | 86.13 | 60.73 | 69.30 | 56.58 | 85.40 | 81.39 | 85.05 |
| GloVe | 84.15 | 61.34 | 68.90 | 68.75 | 82.32 | 79.44 | 83.20 |
| FastText | 88.60 | 77.80 | 83.00 | 76.70 | 87.70 | 84.80 | 86.57 |
| Word2Vec | 85.42 | 65.81 | 70.27 | 70.65 | 85.13 | 83.92 | 84.36 |
| BERT | 88.15 | 76.30 | 84.80 | 77.01 | 86.34 | 85.23 | 86.34 |
| MBERT | 88.53 | 76.45 | 84.61 | 76.50 | 85.48 | 84.64 | 85.93 |
| DBERT | 90.26 | 79.02 | 85.95 | 78.84 | 88.92 | 87.15 | 87.27 |
| RoBERTa | 89.38 | 77.39 | 84.56 | 77.42 | 88.19 | 86.71 | 86.80 |

| Table 4. The classification accuracies of all techniques without resampling at ts50. |
|---|---|---|---|---|---|---|---|
| Models | Hitman | Soccer Manager | GTA San Andreas | City Driving | Amazon | AllExpress | Ebay |
| CNN | 83.45 | 69.80 | 74.17 | 70.61 | 81.50 | 78.49 | 80.32 |
| RNN | 84.43 | 66.94 | 69.34 | 69.31 | 82.75 | 76.84 | 80.61 |
| LSTM | 83.69 | 58.60 | 65.56 | 52.25 | 82.68 | 79.57 | 81.23 |
| GloVe | 80.22 | 57.02 | 63.80 | 60.45 | 78.50 | 75.62 | 77.58 |
| FastText | 84.91 | 73.20 | 78.30 | 71.10 | 82.98 | 80.08 | 79.70 |
| Word2Vec | 82.88 | 62.70 | 69.45 | 67.36 | 79.31 | 77.10 | 75.54 |
| BERT | 85.33 | 73.48 | 80.08 | 73.29 | 83.52 | 81.41 | 82.99 |
| MBERT | 85.71 | 73.63 | 80.89 | 72.78 | 82.66 | 81.82 | 82.11 |
| DBERT | 87.04 | 75.92 | 82.53 | 76.02 | 85.10 | 84.33 | 84.55 |
| RoBERTa | 86.36 | 74.57 | 81.74 | 73.60 | 84.37 | 83.99 | 83.07 |
78.61% of accuracy among word embedding models. Furthermore, decrement of training set percentage is disadvantageous in terms of decreasing the average classification success of LSTM from 74.94% of accuracy to 71.95% of accuracy. The reduction of training set size emerges that RNN performs worse classification success with nearly 5% decrement in average classification accuracy. When the classification performance of RNN decreases approximately 5%, the LSTM also deteriorates the system performance with roughly 3% decrement. When other techniques are taken into account, there is no significant decrease is observed in classification accuracy at ts50. Whilst deep contextualized word representations are also included, DBERT outperforms both deep learning models and conventional word embedding methods with 82.21% of mean accuracy. On the other hand, GloVe exposes the poorest classification performance with 70.46% of average accuracy. As with ts80, BERT-related models demonstrate the best classification performance at ts50 by performing nearly 80% of mean accuracy and above. Another observation from Table 4 is the success order of deep contextualized word representations similar to ts80 as follows: DBERT > RoBERTa > BERT > MBERT. As a result of Tables 3 and 4, all algorithms show classification performance of 80% and above in datasets that have a higher number of documents (Hitman, Amazon, AliExpress, Ebay) than the others. This, in turn, allows us to conveniently observe the effect of the number of documents on classification performance.

In Table 5, the classification accuracies of all methods are demonstrated at ts30. The performance order is observed among deep learning techniques at ts30 as follows: CNN > RNN > LSTM while the classification success of word embedding models is ordered as: FastText > Word2Vec > GloVe. When average sentiment classification accuracies of the users are considered, the overall performances of all models are obtained as DBERT > RoBERTa > BERT > MBERT > FastText > CNN > RNN > Word2Vec > LSTM > GloVe. In Table 5, similar to the results of Tables 3 and 4, DBERT presents the best classification performance while GloVe demonstrates the poorest success. As a result of Tables 3–5, the classification performances of successful methods among deep learning and word embedding models namely, FastText and CNN demonstrate very similar accuracy values in every training set size. This means that the change of training set percentages does not have any significant effect on the classification performance of CNN and FastText models. To sum up, the selection of the DBERT model for the purpose of categorizing the sentiment of users on each mobile application boosts the system performance at every training set size.

Table 5. The classification accuracies of all techniques without resampling at ts30.

| Models   | Hitman | Soccer Manager | GTA San Andreas | City Driving | Amazon | AliExpress | Ebay |
|----------|--------|----------------|-----------------|--------------|--------|------------|------|
| CNN      | 70.70  | 58.58          | 62.51           | 58.14        | 69.07  | 67.36      | 71.00|
| RNN      | 70.78  | 55.25          | 58.44           | 56.71        | 70.59  | 65.12      | 70.22|
| LSTM     | 71.94  | 49.94          | 56.80           | 41.58        | 59.88  | 68.35      | 71.38|
| GloVe    | 67.50  | 47.37          | 53.64           | 47.92        | 65.09  | 64.80      | 67.85|
| FastText | 71.26  | 61.55          | 65.70           | 59.85        | 69.76  | 69.00      | 69.04|
| Word2Vec | 65.13  | 52.05          | 58.11           | 57.40        | 66.43  | 65.25      | 65.29|
| BERT     | 74.61  | 63.73          | 68.00           | 61.22        | 72.70  | 71.42      | 72.28|
| MBERT    | 74.08  | 63.09          | 68.52           | 60.93        | 71.12  | 71.26      | 72.64|
| DBERT    | 76.31  | 65.25          | 71.75           | 63.48        | 74.01  | 73.95      | 73.25|
| RoBERTa  | 75.74  | 64.82          | 70.63           | 62.25        | 73.13  | 72.40      | 72.45|
The detailed statistics of the datasets before data sampling is shown in Table 1. After applying random under sampling (RUS) and word embedding substitution (WES) techniques to balance the training data sets, statistics of the datasets are given in Table 6. In Table 6, number of training set instances is presented for training set 80%. In order to eliminate skew data distribution in positive and negative classes, random under sampling and word embedding substitution methods are used. In all datasets, the RUS method is used to balance the number of positive comments with the number of negative comments, since the number of positive comments is greater than the number of negative ones. In terms of equalizing the number of negative and positive comments with each other, the WES method is preferred to reproduce the negative comments. Thus, more meaningful instances are obtained instead of randomly reproducing negative samples that are a minority.

In Table 7, the classification accuracies of all techniques are demonstrated using RUS and WES sampling techniques at ts80 in order to avoid data imbalance. It is clearly seen that sampling techniques affect classification accuracies positively for all methods on Hitman, AliExpress, Ebay, and Amazon datasets. Enhancement in classification is provided by sampling methods nearly 3–4% for Hitman, about 2–3% for AliExpress and Ebay, and approximately 1–2% for Amazon datasets. On the other hand, data sampling techniques exhibit negative effect by observing almost 1–2% decrement in classification performance on Soccer Manager, GTA San Andreas, City Driving datasets when compared to without applying resampling techniques at ts80. We consider that the reason behind the poor classification performance of data sampling techniques in these three data sets is because the class distributions are closer compared to the other datasets. For this reason, we observe that data sampling techniques further reduce classification success than in its original version. On the other hand, the reason for the increase in classification success of data sampling techniques in other data sets is skewed data distribution in

| Dataset       | #Positive | #Negative | Total   |
|---------------|-----------|-----------|---------|
| Hitman        | 8310      | 8310      | 16,620  |
| Soccer Manager| 2284      | 2284      | 4568    |
| GTA San Andreas| 1938      | 1938      | 3876    |
| City Driving  | 4284      | 4284      | 8568    |
| Amazon        | 23,022    | 23,022    | 46,044  |
| AliExpress    | 28,834    | 28,834    | 57,668  |
| Ebay          | 27,840    | 27,840    | 55,680  |

| Models | Hitman | Soccer Manager | GTA San Andreas | City Driving | Amazon | AliExpress | Ebay   |
|--------|--------|----------------|-----------------|--------------|--------|------------|--------|
| CNN    | 89.12  | 71.92          | 76.70           | 74.85        | 86.32  | 85.30      | 86.44  |
| RNN    | 89.73  | 71.10          | 74.06           | 75.68        | 88.57  | 83.80      | 88.25  |
| LSTM   | 89.35  | 59.82          | 68.63           | 54.44        | 86.40  | 83.55      | 87.92  |
| GloVe  | 86.42  | 61.05          | 66.12           | 67.53        | 83.32  | 81.13      | 85.60  |
| FastText| 90.88  | 76.54          | 81.28           | 74.02        | 88.70  | 86.35      | 88.64  |
| Word2Vec| 86.75  | 64.36          | 69.94           | 69.60        | 87.13  | 85.81      | 86.36  |
| BERT   | 90.49  | 75.95          | 83.60           | 76.72        | 88.34  | 87.29      | 88.14  |
| MBERT  | 91.80  | 75.70          | 82.48           | 75.32        | 86.48  | 87.72      | 87.39  |
| DBERT  | 93.01  | 78.66          | 84.38           | 77.18        | 90.92  | 89.45      | 89.05  |
| RoBERTa| 92.53  | 77.00          | 83.31           | 76.80        | 89.19  | 88.05      | 88.62  |
positive and negative classes. Thus, in datasets where class distributions are unbalanced, as expected, data sampling techniques significantly improve classification performance.

As a result of Table 7, mean classification accuracies for determining the sentiment of users is obtained in the following order DBERT > RoBERTa > BERT > MBERT > FastText > RNN > CNN > Word2Vec > GloVe > LSTM. This means that DBERT outperforms others with 86.09% of accuracy. It is followed by RoBERTa with 85.07% of accuracy, BERT with 84.36% of accuracy, MBERT with 84.84% of accuracy, FastText with 83.77% of accuracy, RNN with 81.60% of accuracy, CNN with 81.52% of accuracy, Word2Vec with 78.56% of accuracy, GloVe with 75.88% of accuracy, LSTM with 75.73% of accuracy. When the average accuracy results of Tables 3 and 5 are compared, the usage of data sampling techniques for classification models generally boosts the classification success with nearly 1% improvement.

In Table 8, the classification results of different evaluation metrics such as accuracy, F-measure, precision, sensitivity, and Matthew’s correlation coefficient (MCC) are presented for the DBERT model. Precision quantifies the number of positive class predictions that actually belong to the positive class, while recall quantifies the number of positive class predictions made out of all positive examples in the dataset. Thus, F-measure is the harmonic mean of the precision and recall while sensitivity measures the proportion of positives that are correctly identified. MCC is another evaluation metric that is actually a correlation coefficient between the observed and predicted binary classifications; it returns a value between −1 and +1. A coefficient of ‘+1’ demonstrates a perfect prediction while ‘−1’ shows total disagreement between estimation and observation. When MCC values are evaluated in Table 8, it is obviously observed that DBERT model is quite capable and consistent to classify sentiment of users.

5. Discussion

In this study, we focus on determining the loyalty prediction of users on different mobile applications by evaluating sentiment analysis of users. In order to analyse the tendency of users, the sentiment of users is appraised by deep learning techniques, word embedding models, and deep contextualized word representations. Convolutional neural networks (CNNs), recurrent neural networks (RNNs), long short-term memory networks (LSTMs) are employed as deep learning techniques, while Word2Vec, GloVe, and FastText are used as word embedding models. Furthermore, bidirectional encoder representations from transformers (BERT), Multilingual BERT (MBERT), DistilBert (DBERT), A Robustly Optimized BERT (RoBERTa) are evaluated as deep contextualized word representations in the study.

Table 8. Statistics of the training datasets after using RUS and WES sampling methods at ts80.

| Dataset       | Accuracy | F-measure | Precision | Sensitivity | MCC  |
|---------------|----------|-----------|-----------|-------------|------|
| Hitman        | 93.01    | 96.14     | 94.35     | 93.20       | 0.48 |
| Soccer Manager| 78.66    | 74.83     | 77.70     | 72.98       | 0.55 |
| GTA San Andreas| 84.38   | 82.61     | 81.50     | 79.94       | 0.38 |
| City Driving  | 77.18    | 74.44     | 75.87     | 75.30       | 0.52 |
| Amazon        | 90.92    | 91.57     | 92.25     | 90.34       | 0.46 |
| AliExpress    | 89.45    | 92.55     | 90.34     | 89.13       | 0.41 |
| Ebay          | 89.05    | 91.63     | 89.94     | 93.56       | 0.36 |
To our knowledge, this is the first attempt to evaluate the loyalty tendency of customers by analyzing sentiments of users from their comments on mobile applications using deep learning techniques, word embedding models, and deep contextualized word representations. In order to show the effectiveness of the proposed model, experiments are carried out on seven different datasets. DBERT model exhibits the best sentiment classification success with 86.09% of mean accuracy value at ts80. Considering that the bidirectionality, ability of the combination of mask language model and next sentence prediction and understanding context-heavy texts, the classification performance of BERT-related models outperform the others. As expected, DBERT exhibits the best classification performance among BERT-related models because of decreasing the size of a BERT model by 40%, while keeping 97% of its language comprehension capacity and being 60% faster.

In the study (Kilimci et al., 2020), authors concentrate on the churn estimation in mobile games using word embedding models and deep learning algorithms. They report that FastText model exhibits the best sentiment classification success with 81.52% accuracy value at ts80. In this work which is the extended version of the study (Kilimci et al., 2020), we improve the classification performance by providing nearly 4% enhancement in the mean accuracy at the same conditions with inclusion of DBERT. Unlike our previous study (Kilimci et al., 2020), random under sampling and word embedding substitution techniques are included in order to eliminate the class imbalance, and the system contribution of the DBERT model increased to 5%. Moreover, BERT-related models also boost the classification performance of the system by outperforming the best model of our previous study (FastText) (Kilimci et al., 2020) about minimum 2% and almost maximum 4% advancement when sampling techniques are applied at ts80. This means that the usage of BERT-related models and inclusion of data sampling techniques facilitate more successful prediction models to analyse the sentiments of users which can evaluate the loyalty of users. Thus, the loyalty estimation of users can be foreseen in the early stages by analyzing most of the users’ sentiments if there is a negative side related to the application. This facilitates to take precautions against the churn of users in terms of the business side and application developers.

To sum up, the trained models determine the positive or negative approach of the user to the application by evaluating user loyalty according to the application. Among the trained models employed in this context, the DBERT model is the most successful method for detecting positive or negative comments of the same users, with an average accuracy of 86.09%. The results of the experiment showed that the same users, especially those who commented on the mobile application three and four times, reported their positive comments about the application and their requests to improve the application. Thus, we conclude that there is a connection between the sentiment analysis of customers from their feedbacks and customer loyalty. In conclusion, the experiment results show that the sentiment analysis of users in mobile applications can be a powerful indicator in terms of predicting customer loyalty. In future, we plan to build a hybrid loyalty prediction system based on both analysing sentiments of users and demographic, economic, or behavioural data about their users on mobile applications.
Notes

1. https://play.google.com/store/apps/details?id=com.squareenixmontreal.
hitmansniperandroid&hl=tr
2. https://play.google.com/store/apps/details?id=com.soccermanagerltd.worlds&referrer=utm_source%3Dweb%26utm_medium%3Dweb%26utm_campaign%3Ddesktop-web-index
3. https://play.google.com/store/apps/details?id=com.rockstargames.gtasa&hl=tr
4. https://play.google.com/store/apps/details?id=com.zuuks.city.driving&hl=tr
5. https://play.google.com/store/apps/details?id=com.amazon.mShop.android.shopping&hl=tr
6. https://play.google.com/store/apps/details?id=com.alibaba.aliexpresshd&hl=tr
7. https://play.google.com/store/apps/details?id=com.ebay.mobile&hl=tr

Acknowledgements

This study is extended from International Symposium on INnovations in Intelligent SysTems and Applications (INISTA 2020). Zeynep Hilal Kilimci contributes the extended version of the conference study in INISTA 2020.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

The author(s) reported there is no funding associated with the work featured in this article.

Notes on Contributor

Zeynep Hilal Kilimci graduated from the Computer Engineering Department, Dogus University, in 2008, the M.Sc. degree from the Computer Engineering Department, Dogus University, in 2013, and the Ph.D. degree from the Computer Engineering Department, Kocaeli University, in 2018. From 2009 to 2011, she worked as a Software Engineer in CRM Department and the Data Warehouse Department in Deniz Bank. In 2011, she started to work as a Research Assistant at the Computer Engineering Department in Dogus University, where she also worked as an Assistant Professor, between 2018 and 2020. She is currently working as an Assistant Professor at Kocaeli University. Her research fields include text mining, data mining, machine learning, ensemble learning, deep learning, reinforcement learning, and artificial intelligence. She serves as a scientific referee for more than 20 journals such as Knowledge-Based Systems, Expert Systems with Applications, Journal of Experimental & Theoretical Artificial Intelligence, Soft Computing, IEEE Transactions on Cybernetics, IEEE Access, IEEE Transactions on Knowledge and Data Engineering.

ORCID

Zeynep Hilal Kilimci http://orcid.org/0000-0003-1497-305X

References

Bengio, Y., Ducharme, R., Vincent, P., & Janvin, C. (2003). A neural probabilistic language model. The Journal of Machine Learning Research, 3, 1137–1155.
Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics, 5, 135–146. https://doi.org/10.1162/tacl_a_00051

Bruns, A., Kornstadt, A., & Wichmann, D. (2009). Web application tests with selenium. IEEE Software, 26(5), 88–91. https://doi.org/10.1109/MS.2009.144

Buckinx, W., Verstraeten, G., & Van den Poel, D. (2007). Predicting customer loyalty using the internal transactional database. Expert Systems with Applications, 32(1), 125–134. https://doi.org/10.1016/j.eswa.2005.11.004

Castro, E. G., & Tsuzuki, M. S. (2015). Churn prediction in online games using players’ login records: A frequency analysis approach. IEEE Transactions on Computational Intelligence and AI in Games, 7(3), 255–265. https://doi.org/10.1109/TCIAIG.2015.2401979

Collobert, R., & Weston, J. (2008, July). A unified architecture for natural language processing: Deep neural networks with multitask learning. In Proceedings of the 25th international conference on machine learning (pp. 160-167).

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv Preprint ArXiv:1810.04805. 1-16.

Elman, J. L. (1990). Finding structure in time. Cognitive Science, 14(2), 179–211. https://doi.org/10.1207/s15516709cog1402_1

Graves, A., Mohamed, A. R., & Hinton, G. (2013, May 26-31). Speech recognition with deep recurrent neural networks. In 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (pp. 6645-6649). Ieee.

Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. (2016). LSTM: A search space odyssey. IEEE Transactions on Neural Networks and Learning Systems, 28(10), 2222–2232. https://doi.org/10.1109/TNNLS.2016.2582924

Hadjii, F., Sifa, R., Drachen, A., Thurau, C., Kersting, K., & Bauckhage, C. (2014, August 26-29). Predicting player churn in the wild. In 2014 IEEE Conference on Computational Intelligence and Games (pp. 1-8). IEEE.

Hallowell, R. (1996). The relationships of customer satisfaction, customer loyalty, and profitability: An empirical study. International Journal of Service Industry management, 4(4).

Huang, B., Kechadi, M. T., & Buckley, B. (2012). Customer churn prediction in telecommunications. Expert Systems with Applications, 39(1), 1414–1425. https://doi.org/10.1016/j.eswa.2011.08.024

Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2016). Bag of tricks for efficient text classification. arXiv Preprint ArXiv:1607.01759. 1-5.

Kawale, J., Pal, A., & Srivastava, J. (2009, August 29-31). Churn prediction in MMORPGs: A social influence based approach. In 2009 International Conference on Computational Science and Engineering (Vol. 4, pp. 423-428). IEEE.

Kent, D., & Salem, F. (2019, August 4-7). Performance of three slim variants of the long short-term memory (LSTM) layer. In 2019 IEEE 62nd International midwest Symposium on circuits and Systems (MWSCAS) (pp. 307-310). IEEE.

Kilimci, Z. H., Yörük, H., & Akyokus, S. (2020, August 24-26). Sentiment analysis based churn prediction in mobile games using word embedding models and deep learning algorithms. In 2020 International Conference on INnovations in Intelligent SysTemS and Applications (INISTA) (pp. 1-7). IEEE.

Kim, S., Choi, D., Lee, E., & Rhee, W. (2017). Churn prediction of mobile and online casual games using play log data. PloS One, 12(7), 1–19. https://doi.org/10.1371/journal.pone.0180735

Kim, Y. (2014, October 25–29). Convolutional neural networks for sentence classification. In 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 1746–1751). Association for Computational Linguistics.

Kristensen, J. T., & Burelli, P. (2019, August 20-23). Combining sequential and aggregated data for churn prediction in casual freemium games. In 2019 IEEE Conference on Games (CoG) (pp. 1-8). IEEE.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. Communications of the ACM, 60(6), 84–90. https://doi.org/10.1145/3065386
Kumar, A., & Chandrakala, D. (2016). A survey on customer churn prediction using machine learning techniques. *International Journal of Computer Applications*, 154(10), 13–16. https://doi.org/10.1145/3065386

Le, Q., & Mikolov, T. (2014, June 21-26). Distributed representations of sentences and documents. In International Conference on Machine Learning (pp. 1188-1196). PMLR.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, 521(7553), 436–444. https://doi.org/10.1038/nature14539

Lipton, Z. C., Berkowitz, J., & Elkan, C. (2015). A critical review of recurrent neural networks for sequence learning. *arXiv Preprint ArXiv:1506.00019*. 1-38.

Liu, X., Xie, M., Wen, X., Chen, R., Ge, Y., Duffield, N., & Wang, N. (2018, November 17-20). A semi-supervised and inductive embedding model for churn prediction of large-scale mobile games. In 2018 IEEE International Conference on Data Mining (ICDM) (pp. 277-286). IEEE.

Liu, X., Xie, M., Wen, X., Chen, R., Ge, Y., Duffield, N., & Wang, N. (2020). Micro-and macro-level churn analysis of large-scale mobile games. *Knowledge and Information Systems*, 62(4), 1465–1496. https://doi.org/10.1007/s10115-019-01394-7

Mikolov, T. (2013). Efficient estimation of word representations in vector space. *arXiv Preprint ArXiv:1301.3781*. 1-12.

Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., & Joulin, A. (2017). Advances in pre-training distributed word representations. *arXiv Preprint ArXiv:1712.09405*. 1-4.

Periáñez, Á., Saas, A., Guitart, A., & Magne, C. (2016, October 17-19). Churn prediction in mobile social games: Towards a complete assessment using survival ensembles. In 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA) (pp. 564-573). IEEE.

Pires, T., Schlinger, E., & Garrette, D. (2019). How multilingual is multilingual BERT? *arXiv Preprint ArXiv:1906.01502*. 1-6.

Rahim, A. G., Ignatius, I. U., & Adeoti, O. E. (2012). Is customer satisfaction an indicator of customer loyalty? *International Research Journal of Engineering and Technology*, 5(11), 1-8.

Santharam, A., & Krishnan, S. B. (2018). Survey on customer churn prediction techniques. *International Research Journal of Engineering and Technology*, 5(11), 1-8.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61, 85–117. https://doi.org/10.1016/j.neunet.2014.09.003
Sulistiani, H., & Tjahyanto, A. (2017). Comparative analysis of feature selection method to predict customer loyalty. *IPTEK The Journal of Engineering, 3*(1), 1–5. https://doi.org/10.12962/joe.v3i1.2257

Tamassia, M., Raff, W., Sifa, R., Drachen, A., Zambetta, F., & Hitchens, M. (2016, September 20-24). Predicting player churn in destiny: A hidden Markov models approach to predicting player departure in a major online game. In 2016 IEEE Conference on Computational Intelligence and Games (CIG) (pp. 1-8). IEEE.

Viljanen, M., Airola, A., Heikkonen, J., & Pahikkala, T. (2017). Playtime measurement with survival analysis. *IEEE Transactions on Games, 10*(2), 128–138. https://doi.org/10.1109/TCIAIG.2017.2727642

Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). Deep learning for computer vision: A brief review. *Computational Intelligence and Neuroscience, 2018*, 1–13. https://doi.org/10.1155/2018/7068349

Wassouf, W. N., Alkhatib, R., Salloum, K., & Balloul, S. (2020). Predictive analytics using big data for increased customer loyalty: Syriatel telecom company case study. *Journal of Big Data, 7*(1), 1–24. https://doi.org/10.1186/s40537-020-00290-0

Wijaya, A., & Girsang, A. S. (2016). Use of data mining for prediction of customer loyalty. *Communication and Information Technology Journal, 10*(1), 41–47. https://doi.org/10.21512/commit.v10i1.1660

Xie, H., Devlin, S., & Kudenko, D. (2016, October 8-12). Predicting disengagement in free-to-play games with highly biased data. In Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (Vol. 12, No. 1).

Xie, H., Devlin, S., Kudenko, D., & Cowling, P. (2015, Aug 31-Sep 2). Predicting player disengagement and first purchase with event-frequency based data representation. In 2015 IEEE Conference on Computational Intelligence and Games (CIG) (pp. 230-237). IEEE.

Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level convolutional networks for text classification. *Advances in Neural Information Processing Systems, 28*(1), 649–657.