Using Word2Vec Recommendation for Improved Purchase Prediction

Ramazan Esmeli
School of Computing
University of Portsmouth
Portsmouth, UK
ramazan.esmeli@port.ac.uk

Mohamed Bader-El-Den
School of Computing
University of Portsmouth
Portsmouth, UK
mohamed.bader@port.ac.uk

Hassana Abdullahi
School of Mathematics and Physics
University of Portsmouth
Portsmouth, UK
hassana.abdullahi@port.ac.uk

***Abstract*—Purchase prediction can help e-commerce planners plan their stock and personalised offers. Word2Vec is a well-known method to explore word relations in sentences for sentiment analysing by creating vector representation of words. Word2Vec models are used in many works for product recommendations. In this paper, we analyse the effect of item similarities in the sessions in purchase prediction performance. We choose the items from different position of the session, and we derive recommendations from selected items using Word2Vec model. We assess the similarities between items by analysing the number of common recommendations of selected items. We train classification algorithms after we include similarity calculations of the selected items as session features. Computational experiments show that using similarity values of the interacted items in the session improves the performance of purchase prediction in terms of F1 score.**

***Index Terms*—Purchase Intent, Word2vec Product Recommendation, Purchase behaviour prediction, browsing behaviour, Classification, Machine Learning**

I. Introduction

User behaviour in e-commerce platforms provides valuable information about users' preferences and intentions. Analysing user behaviour can help both businesses and e-shoppers. For businesses, they can increase their revenue when they develop strategies based on analysed user behaviour data. For example, using purchase prediction, e-commerce strategist can plan their offers for the user's next session to aid a quick purchase decision. Also, based on the purchase prediction result, the items in the previous session can guide Recommender Systems (RS) [39] to increase content personalisation when next the user visits the website. For e-shoppers, when they get offers and discounts on products that they are interested in, they feel special, which will increase customer satisfaction and loyalty.

RS are used in many domains to filter relevant products for users in online platforms, including movie [20], e-commerce [4], and music [22]. There are many kinds of methods to develop RS with the most commonly used methods being Content-Based RS (CBRS) [40], Collaborative Filtering (CFRS) [42], and Hybrid-Based RS. In CBRS, the attributes of the item and user are used to create the similarities between other items and users. While in CFRS, only user-item interaction history is used. In Hybrid-Based RS, both methods are incorporated. Recently, in Session-Based RS (SBRS), recommendations are produced based on the interacted items in the session, and there is no need to know about previous interactions of the user. Word2Vec [7] is mainly used in sentiment analysis in paragraphs and documents. In Word2Vec, words in the document are represented as vectors. Vectors can have different dimensions, and a word vector is built from the context of the word. Word2Vec is also adopted for top-n product recommendations in several works [30], [31] which show promising results.

Purchase prediction [26] is a remarkable machine learning classification model that can help e-commerce strategists to perform correct actions for their future plans such as stock control. Also, analysing user behaviour can help indicate whether there will be purchase in the next session or not [24]. Purchase prediction has been combined with RS to investigate if there will be purchase in the session and to generate recommendations based on interacted items in the session to predict which item can be purchased [9], [35].

In this paper, we integrate Word2Vec based recommendations with session features to have more robust prediction models. We create item recommendations from Word2Vec model using interacted items which are located in different positions in a session. We calculate the similarities between these items by looking at the number of common derived recommendations using these items. We use calculated similarities as features of the session and purchase prediction models are trained using these session features. Computational experiments show that using item similarities as a session feature increases the performance of prediction models.

The main contributions of this paper are as follows:

1) We develop and implement a framework to use Word2Vec technique to get recommendations and calculate similarities between items in a session using derived recommendations.
2) We integrate calculated similarities between items in the session as features to classification models.
3) We apply different sampling methods to deal with class imbalance problem.
4) We validate the proposed approach on various classification algorithms.

The rest of the paper is organised as follows. Section 2 reviews the use of Word2Vec methods in RS works and works done related to purchase prediction. Section 3 explains the data analysis. Section 4 presents the proposed framework. The experiments, results and discussions are shown in Section 5. The conclusion and proposed some future work directions are presented in Section 6.

II. Related Works

A. Session Logs

Analysing users' behaviours using machine learning techniques has taken the attention of e-commerce planners and researchers. Nowadays, users' spend more time on exploring different products and comparing products in different e-commerce platforms to find the most advantageous one in terms of price and quality [8]. Many well-known e-commerce platforms record users' activities and use this data to have personalised content by giving recommendations [13], [14], and purchase prediction in the sessions [43], [44].

B. Session Based Recommendation

SBRS has a growing trending since their success in providing real-time recommendations even to anonymous users. Many SBRS models are designed[17], [23], [18], [13], [14]. Authors [17] designed RNN based recommendation for short user interaction history their results on Yoochose dataset[1] showed their designed method showed the superiority over the Item-Item similarity-based recommendation. Later, [19] designed a context-aware SBRS that features such as price, category and time of the day were used as a factor of filtering and re-ranking the recommendation list. Their result showed a significant improvement in the recommendation performance in comparison with the base SBRS. [18] designed modified Item-Item similarity-based SBRS and compared the performance with the designed method in [17]. Their comparison results showed that combined Item-Item similarity-based SBRS and RNN based method performed better than single models.

The works on SBRS mainly focused on using deep learning methods, while recent works question the improvement in the performance of designed deep-learning-based SBRS comparing to Item-Item similarity-based SBRS. Also, the applicability of deep learning-based methods. [25] conducted experiments in order to compare the performance of the deep-learning-based SBRS and modified Item-Item similarity-based SBRS. They found that deep learning approaches have still limitations in terms of scalability, running time, complexity and recommendation performance. [12] evaluated recent designed deep learning-based SBRS and compared with less complex Item-Item similarity-based SBRS. Based on their experiment results, most of the deep learning-based approaches cannot be reproduced. Also, they were outperformed by simple SBRS methods. Therefore, in this study, we select simple word2Vec product recommendation model in order to get the most similar products for a given interacted product. However, our approach can work when different SBRS is applied since we use recommendation model in order to calculate the similarities between the items in the session.

[1] https://2015.recsyschallenge.com/challenge.html

C. Word2Vec Recommendation

Word embedding is used to represent words as vectors that describe the word based on its context, such as surrounding words in the sentence. There are two main methods for word embedding with word2Vec method[36], which are skip-gram and the Continuous Bag of Words (CBOW). CBOW can predict the word by using its context; for example, from a given sequence of words, the next word can be predicted(Fig 1).

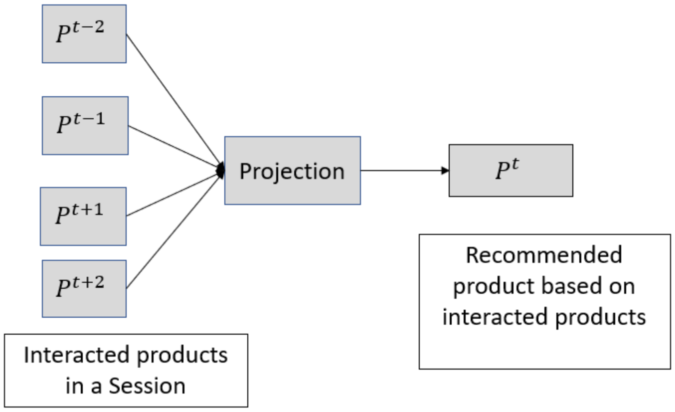
P^{t-2}
P^{t-1}
P^{t+1}
P^{t+2}
Projection
P^t
Recommended product based on interacted products
Interacted products in a Session

Fig. 1. An illustration of the CBOW model for product recommendation

While skip-gram can predict the context using the word, in which based on a given the word, surrounding words which share similar context to the word can be predicted(Fig 2).

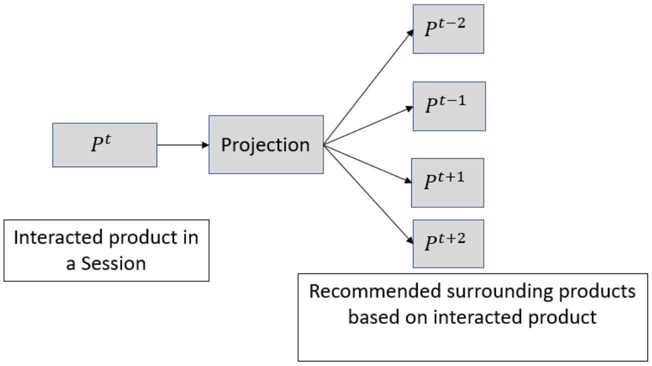
P^t
Projection
P^{t-2}
P^{t-1}
P^{t+1}
P^{t+2}
Interacted product in a Session
Recommended surrounding products based on interacted product

Fig. 2. An illustration of the N-skip-gram model for product recommendation

Word2Vec methods are commonly used for sentiment analysing; for example, in order to examine whether a customer is happy after using a product, this can be determined from feedback using Word2Vec model [43], [41]. The details and parameters of word embedding (Word2Vec) can be found in [27], [36].

Furthermore, modified Word2Vec methods are used in item recommendations in many works [5], [15], [10]. In Word2Vec recommendation, products of basket can be seen as the words of a sentence [5]. Thus, product (item) and word can be accepted as interchangeable. Word2Vec methods can help to represent items as a vector, and vectors can be used to calculate item similarities. After obtaining item similarities, Item-Item recommendations can be provided. In this work, instead of having word embedding, we create product embedding from the sessions, in which each session is a context (sentence). Our Word2Vec RS has a similar approach to [5]. However, we use Word2Vec recommendations to calculate the similarities between items. Finally, we use the calculated similarity as a feature for classification algorithms.

D. *Purchase Prediction*

There are several works that focused on purchase prediction by using different methods. Purchase prediction models depend on the feature engineering, including temporal features and session-based features [8], [6], [24]. On the other hand, in some works, features are learned as a consequence of the click actions using Recurrent Neural Network (RNN) methods [37], [38]. Moreover, in [32], heuristic methods were utilised to have quick coverage by estimating the initial seeds for prediction models. Experiment results show that they were able to get 99% accuracy on purchase intention detection in sessions. However, using the accuracy as the evaluation metric in the imbalanced dataset could mislead the performance of the classification models[28], [1], [2]. In [26], they investigated purchase prediction for the non-contractual setting. They build machine learning models to predict user's intention in the session using extracted features which depend on previously purchasing ended sessions for the same customer. Experiment results showed that their models could reach 88.9% ROC score on predicting users' intention in the sessions.

[29] examined whether a user session will end a purchase or not. In their work, they used a product's temporal features and session features to build the prediction models. In item temporal feature, they added product trendiness over time. They compared the improvement in the prediction model performance with trendiness and without trendiness. [29] is very similar to our paper however in our work, we compare the performance improvement of purchase prediction model not for item trendiness, but in addition to temporal features and session features, we add product similarity scores and investigate the effect of using product similarity scores. Product similarity scores are calculated using Word2Vec RS, in which similarity scores depend on the number of the commonly recommended items for the selected items.

E. *Class Imbalance Problem*

Since purchase action infrequently happens in e-commerce platform, as most of the sessions are non-purchased sessions, the sessions are mostly labelled as non-purchase. Therefore, the non-purchase class will dominate the purchase ended class. Classification algorithms are heavily affected by an imbalanced dataset. In order to reduce this imbalanced class drawback, different methods have been applied, including oversampling [21] and Under-sampling [16]. In this work, we apply the Synthetic Minority Over-sampling Technique (SMOTE) and Under-sampling methods.

III. Dataset Analysing

We use RetailRocket dataset publicly available at [2]to test our proposed framework. The dataset details are given in Table I. It can be seen from the Table I that majority of the sessions end with view only. Therefore, the dataset has substantial class imbalance. To have reliable classification models, we need to apply class imbalance approaches. Moreover, we analysed the dataset in terms of the distribution of interaction types for weekdays over five months (Fig 3). Interestingly, weekdays has more add to cart and purchase events comparing to weekends. Fig 4 shows session frequency distribution in the dataset for top 10 sessions. It is seen that the highest number of item interaction in a session is slightly above 400. Also, we analyse the event type distributions in the sessions. As seen in Figure 5, most of the sessions end with browsing only.

TABLE I
Dataset Statistics before and after pre-processing

dataset	#view	#add to cart	#transaction
Before Pre-processing	2664312	69332	22457
After Pre-processing	1030630	53235	19856

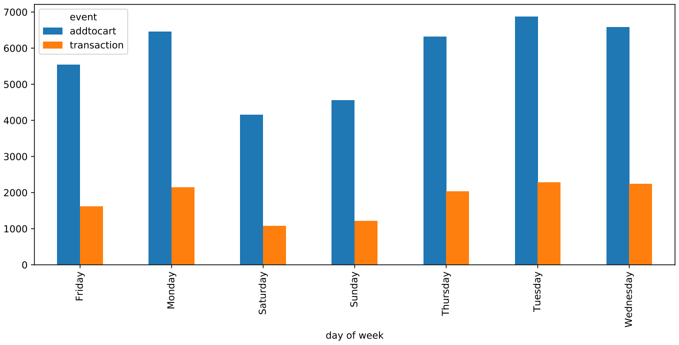

Fig. 3. Frequency distribution of event types for weekdays

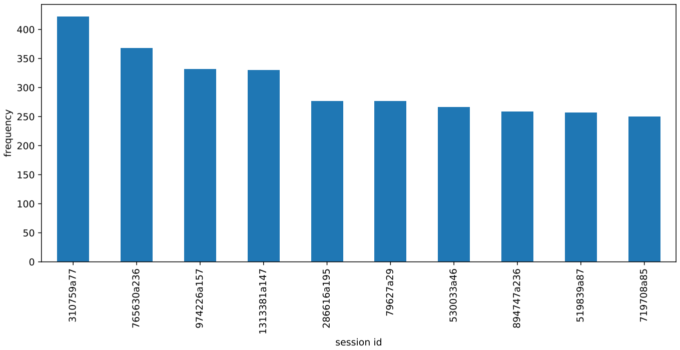

Fig. 4. Session-Item frequency distribution

[2] https://www.kaggle.com/retailrocket/ecommerce-dataset

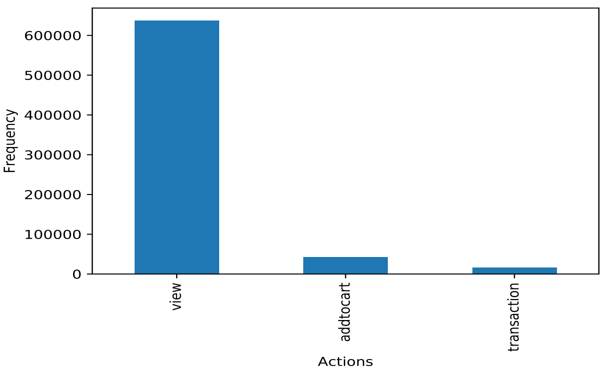

Fig. 5. Frequency distribution of event types in the sessions

IV. Proposed Word2Vec Item Similarity Integration Framework

The proposed framework consist of 4 phases(Fig. 6). In this section these phases are explained.

A. Phase 1

In this phase, the session logs are collected, and data pre-processing is applied. In data pre-processing, session records which have less than three interacted items are eliminated due to testing our method. In order to test our approach, we need to have at least three items in the sessions since we investigate the effect of the using similarity scores of the products which are in three different positions of the session on purchase intention prediction. Also, sessions which have a very short duration such as 2 seconds are filtered out since these sessions will not give sufficient information about users' intention. After elimination new details about dataset are seen in Table I.

B. Phase 2

This phase consists of training Word2Vec model and adding similarity measurements of the items to features.

1) Word2Vec Model Training: Word2Vec model can learn relations of the words from any given document. In session logs, since we do not have any text explanations of the products, we create word sequences using product ids (Fig. 7). The created sequence of product ids for each session represents the session as a sentence. After having sentences, we train the Word2Vec model (Algorithm 1). We selected N-skip-gram method of Word2Vec model to get recommendations since we need the surrounded products of the given product.

2) Similarity Calculation Using Word2Vec: We create similarity measurements using next product recommendations. Recommendations are generated using top n=100 similar products to the given product. We select n=100 due to ignoring less relevant products to the given product in order to calculate more precise similarity between the products in different positions of the session. For example, i_1 is the first interacted item, L_{i_1}=$R(i_1)$ is the recommendation list from this interaction and i_n is the last interacted item and recommendation from this interaction L_{i_n}=$R(i_n)$. So, similarity between items i_1 and i_2 is $S_{(i_1,i_2)} = \frac{|L_{i_1} \cap L_{i_2}|}{|L_{i_1} \cup L_{i_2}|}$. The similarity calculation of items in different positions in the session are used as session features (Fig. 8). The results of classification models are explained in the experiment and results section.

Algorithm 1 The algorithm of Word2Vec based recommendation integration to purchase prediction

```
$Train$                                   // sessions in Train dataset
$Test$                                    // sessions in Test dataset
$Model_{Word2vec}$                        // Word2vec model
classification_Model                      // Classification model
$sentences$=$Create\_sentence\_representation(Train)$
// a session is a sentence, product_id is a word
$Word2vec\_Model(sentences)$    // Train word2vec model using sentences
$CreateFeatures(Dataset)$       // Create feature for all sessions
for each $s$ in $Sessions$ do
    $n = len(s)$     // the number of interacted items in the session
    $i_f = s[0]$                          // first interacted item
    $i_m = s[\frac{n}{2}]$    // interacted item that is in the middle position of the session
    $i_l = s[n]$                          // last interacted item
    $R_f$= $Model_{Word2vec}(i_f)$        // recommendations from $i_f$
    $R_m$= $Model_{Word2vec}(i_m)$        // recommendations from $i_m$
    $R_l$= $Model_{Word2vec}(i_l)$        // recommendations from $i_l$
    $Similarity(i_f, i_m) = \frac{|R_f \cap R_m|}{|R_f \cup R_m|}$   // similarity calculation between $i_f$ and $i_m$
    $Similarity(i_m, i_l) = \frac{|R_m \cap R_l|}{|R_m \cup R_l|}$   // similarity calculation between $i_m$ and $i_l$
    $Similarity(i_f, i_l) = \frac{|R_f \cap R_l|}{|R_f \cup R_l|}$   // similarity calculation between $i_f$ and $i_l$
    Add_Similarities(s)    // add calculated similarities to session $s$ as session features
end for
$classification\_model(Train\_Dataset)$              // train classification model
$evaluate_{classification\_model}(Test\_Dataset)$    // test classification model
```

C. Phase 3

This phase mainly focuses on attribute selection, feature creation and class imbalance problem.

1) Attribute Selection and Feature Generation: Session logs have important attributes that show user intention, such as whether a user will proceed to purchase or not. In this section, we explain the attributes that are selected from session logs and used to create new features from selected attributes.

1) Total clicks: Indicates how many products were browsed regardless of including redundant products or not.
2) Unique items seen: shows the number of different browsed items in the session.
3) Duration: Shows the total time that the session lasted in seconds.

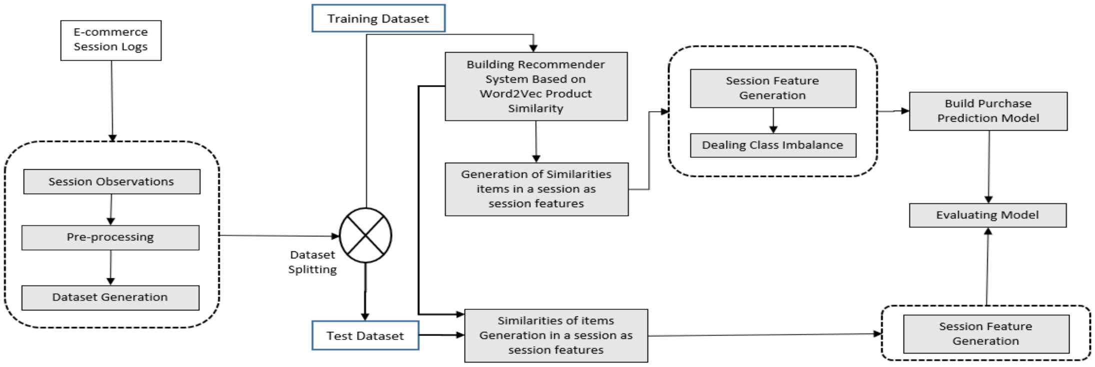
E-commerce Session Logs
Session Observations
Pre-processing
Dataset Generation
Training Dataset
Dataset Splitting
Test Dataset
Building Recommender System Based on Word2Vec Product Similarity
Generation of Similarities items in a session as session features
Similarities of items Generation in a session as session features
Session Feature Generation
Dealing Class Imbalance
Build Purchase Prediction Model
Evaluating Model
Session Feature Generation

Fig. 6. Proposed framework for item similarity to purchase prediction

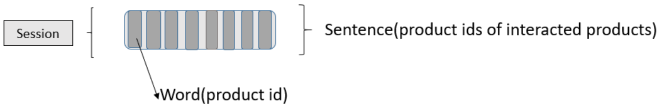

Fig. 7. Sentence representation of product ids in a session

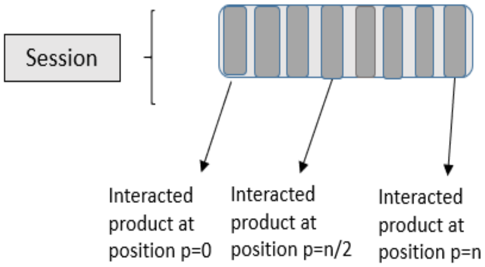
Session
Interacted product at position p=0
Interacted product at position p=n/2
Interacted product at position p=n

Fig. 8. Position of user's interacted items in a session

4) Average duration per item: Shows the total time spent on an item in the session.
5) Month: This feature is extracted from the timestamp attribute that shows the month the session started since users have different habits in different seasons.
6) Hour: Shows the hour of the day the session occurs since some users are more likely to purchase after working hours.
7) Weekday: Shows the day of the week that the session occurred.
8) Weekend: Shows if the session happened on the weekend since dataset analysing shows that most users visit e-commerce websites on the weekend.
9) Similarity: Shows the similarity feature(s) between first-middle, first-last, middle-last or all these similarity scores. These similarity scores are calculated using Word2Vec RS model.

2) Dealing with Class Imbalance Problem: As seen in Table I dataset mainly has non-purchase sessions; the minority of the classes have purchase class. This affects the performance of ML models. Therefore, we apply class imbalance techniques to have a balanced class distribution that can help better ML models. In this work, we apply SMOTE and Under-sampling class imbalance methods and compare the performance results to find out which one is more effective.

D. Phase 4

In this phase, ML models are trained and evaluated. For training and evaluation of the models, we use 10-cross validation strategy. We run experiments to identify the pair of items that create the best F1 score. For each experiment, similarities of items from different positions of the session are included as features of the session. The average F1 score of cross-validation is used as a result of model performance. We use different ML models for classification in the experiments, which are Random Forest (RF)[3], Bagging[11], and Decision Trees (DT)[11]. These models are trained on different class imbalance methods to identify the best method. All the parameters for the classifiers remain as default as shown.

V. Experiments, Results and Discussion

In this section, we explain the design of experiments and the experiment results. We run experiments for two sampling methods. For each sampling method, we executed eight different experiments in order to see the effect of adding similarity scores as features while training the prediction models.

A. Experimental Design

In the experiments, we compare the performance difference when we include similarity attribute to model training under different sampling strategies to deal with the class imbalance problem. In addition, we measure similarities for products which are in different positions in the session. In other words, we measure the similarity between first and last products, first product and product in the middle position of the session, and finally, products in the middle and last positions of the session. The F1 results show the average of the ten cross-validations. In each cross-validation, the product similarities are calculated using only train dataset in order to prevent biased results. We eliminate the session which has less than three interacted items since to calculate the product similarities, the co-occurrences of the products are important. For Word2Vec parameters, we set $vector\ dimension = 100$, $iteration = 30$, $window = 3$, $mincount = 1$ other parameters remained as default. Training of classification models is carried on datasets which have different class imbalance levels (Fig. 9).

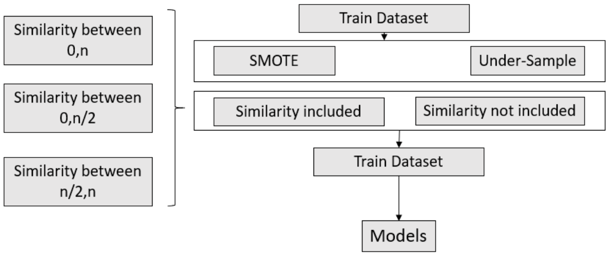

Fig. 9. Design of experiment process

B. Evaluation Metrics

Precision, Recall and F1 are metrics mainly used to measure the performance of classification algorithms. In this study, we use F1 since this metric reflects both Recall and Precision. After cross validation, final F1 metric is calculated as the average of cross validation F1 score.

C. Results

We show the experiment results in Table II. In this Table II, f shows first interacted item, m shows interacted item in the middle position of the session, l shows last interacted item in the session and we compare eight different situations. These situations are , without including any item similarity as features which is called $similarity\ not\ included$, when all calculated similarity score between selected items (f,m,l) called $all\ included$, similarities between f and l called $f - l$, similarities between f and m called $f - m$, similarities between m and l called $m - l$, both $m - l$ and $f - m$, both $f - l$ and $f - m$, and finally both $f - l$ and $m - l$. Also, in Table II, we compare the performance difference when we apply SMOTE and Under-sampling methods to deal with class-imbalance problem.

1) Improvement on Purchase Prediction after Integrating Similarity between Items as Feature: The results show that prediction algorithms produce different performance results when we integrate similarity scores of items in different positions of a session. In Bagging, when both similarities between f and l, and the f and m are included as features, we got the highest F1 score with 88.6 %. However, RF and DT achieved the highest score when we include similarity scores between f and l with 81.8 % and 87.6 % respectively.

Interestingly, although all models give different performance results when we include similarity scores of the items in different positions, the difference between the worst and the best performing scores in the models are close to each other. In Bagging, the F1 score difference between the worst and the best-performed scores are 3.5 %. In RF, this difference is 3.9 %. Lastly, the difference between worst and best score is 3.7 % for DT. Moreover, we investigated the difference in $p \leq 0.05$ in SPSS, applying T-test. The results showed that statistically, there is a significant difference in the models' performance when similarity is included and not included in Bagging, RF, DT with p=0.034,p=0.032, p=0.012, respectively.

Overall, when similarities between items in different positions of the session are not included, the prediction models performed worse compared to when similarity scores are used.

2) SMOTE: Experiment results suggest that SMOTE performs better than Under-sampling in all classification models when dealing with class imbalance (Table II). Also, when we do not include similarity measurements of the viewed products in the session, the performance of classification models is reduced. However, adding all similarity measurements of products which are calculated from products in different positions in the session does not provide the best performing model. When we add all similarity measurements, Bagging model shows the highest performance among other models with 88.6% F1 score. On the other hand, when we apply Under-sampling, RF model gives a better result with 75.1% F1 score.

3) Under-sampling: When similarities are not included, RF is the best performing model with 74% F1 score (Table II). When all similarities gathered from items in different position are integrated as session features, RF and Bagging give a similar F1 score. On the other hand, the RF model outperforms the other models with a 75% F1 score when only the similarity between items that are in the middle and beginning of the session are considered. Nevertheless, overall, computational experiments showed that Under-sampling is not the best way to deal with the class imbalance problem for our dataset.

D. Discussion

Using product similarities as a feature in the session shows better results in predicting users' purchase intention. However, each model reacted differently when the position of the selected products is changed. In the experiments, we used ensemble classification algorithms. As seen from Table II, when we use SMOTE for the imbalanced datasets, all models produce better results in comparison to Under-sampling. Also, Bagging outperformed others when SMOTE is applied. Interestingly, when the classification models are trained on Under-sampled dataset, the RF model gives better prediction result. The other aspect of our work is the positions of selected products for similarity calculation. When similarity values between products in the session are used as session features, classification models show better performance in terms of F1 score. Therefore, product similarities in the session are important indicators to identify user purchase intentions.

Comparing our approach with previous works done on purchase prediction by adding newly created features, [29] analysed the effect of product trendiness as a feature on improving purchase prediction accuracy. They used logistic, Bagging, NBTree and XGBoost classifiers as prediction models. They found that adding the product trendiness as feature improved the performance of the classification models. While [26] applied Gradient tree boosting classifier on unstructured

TABLE II
EFFECT OF INTEGRATING SIMILARITIES OF THE PRODUCTS IN THE SESSION ON PURCHASE PREDICTION PERFORMANCE WHEN DIFFERENT CLASS IMBALANCE METHODS ARE APPLIED

Dataset	Similarity attribute(s)	Bagging(F1)	RF(F1)	DT(F1)
SMOTE	similarity not included	0.851 ± 0.004	0.779 ± 0.005	0.839 ± 0.005
	all included	0.884 ± 0.004	0.807 ± 0.003	0.864 ± 0.003
	f-l	0.883 ± 0.003	0.818 ± 0.004	0.876 ± 0.003
	f-m	0.885 ± 0.003	0.815 ± 0.003	0.868 ± 0.004
	m-l	0.871 ± 0.003	0.809 ± 0.005	0.856 ± 0.003
	m-l,f-m	0.884 ± 0.003	0.802 ± 0.004	0.865 ± 0.003
	f-l, f-m	0.886 ± 0.004	0.805 ± 0.003	0.868 ± 0.003
	f-l,m-l	0.883 ± 0.003	0.803 ± 0.005	0.866 ± 0.004
Under-sampling	similarity not included	0.724 ± 0.006	0.739 ± 0.004	0.694 ± 0.006
	all included	0.743 ± 0.009	0.743 ± 0.006	0.706 ± 0.008
	f-l	0.738 ± 0.007	0.748 ± 0.008	0.703 ± 0.007
	f-m	0.739 ± 0.007	0.745 ± 0.009	0.706 ± 0.008
	m-l	0.735 ± 0.010	0.751 ± 0.006	0.702 ± 0.006
	m-l,f-m	0.744 ± 0.007	0.742 ± 0.005	0.708 ± 0.008
	f-l, f-m	0.740 ± 0.006	0.742 ± 0.005	0.708 ± 0.006
	f-l,m-l	0.742 ± 0.008	0.743 ± 0.005	0.707 ± 0.007

e-commerce dataset. They used the purchase trendiness, customer and item specifications as an attribute of the classification model. Their experiment results showed an 89 % accuracy score. In our work, we used both item feature and users' session feature; also, we included similarities between items in the session as features for classification algorithms. Our experiment results showed 88.6 % F1 score on publicly available dataset[3].

One of our limitations in this work is that we examine our approach only using Word2Vec word embedding method. However, there are other methods such as Gloves [33] and ELMo [34] that could be used for product embedding and calculating the similarities between the products in a session, Also, a different recommender model such as RNN based RS[17] could be utilised in order to get recommendations. Another limitation of our work is that we used only one dataset. Other session-based e-commerce datasets can be used in order to evaluate the robustness of our approach.

VI. CONCLUSION AND FUTURE WORK

Purchase prediction is an important factor for e-commerce decision-makers to give offers and recommendations to the customers. In this paper, we integrated product similarities as a feature of classification models to improve prediction accuracy. We calculated product similarities using Word2Vec method, in which the session represents the sentence, and the product id represents the word that creates the sentence. We chose items in different positions in a session to find the best combination for the best purchase prediction accuracy.

In addition, we used three different ensemble classification models to identify the best performing one. Our experiments showed that the performance of each classification model reacted differently when the selected item's position changes. Also, when interacted product similarities are used as a feature, the accuracy of each model improves. Finally, we experimented with different class imbalance methods to show which method is the most appropriate for imbalance dataset. Computational experiments suggest that Minority Over-sampling (SMOTE) produces better results when compared to Under-sampling method.

Analysing users behaviours in a session may exploit interesting patterns on users' intention of visiting an e-commerce website. In this work, successfully, we showed that considering the pattern of the interacted items similarities in the session can lead to better determination of users' purchase intention. The results of this study may help the e-commerce strategists to give real-time discounts based on the level of the similarities of the items in the session users are interacted, that can convert browsing the products to purchasing the products by convincing the users.

As a future direction, it could be interesting to evaluate the performance of purchase prediction results when integrated with RS to help identify products that can be purchased in a session. Moreover, the similarity calculation between the interacted items can be calculated different approaches, such as using Item-Item similarity and deep-learning-based RS. Also, the diversity of the products in the session can be considered as a new feature as opposed to products similarity. In addition, in a future study, we intend to examine how early purchase intention can be predicted by using similarities between items in the session as a signal and improving the product recommendation with the combination of early purchase intention prediction.

REFERENCES

[1] Bader-El-Den, M.: Self-adaptive heterogeneous random forest. In: 2014 IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA). pp. 640–646. IEEE (2014)

[2] Bader-El-Den, M., Teitei, E., Adda, M.: Hierarchical classification for dealing with the class imbalance problem. In: 2016 International Joint Conference on Neural Networks (IJCNN). pp. 3584–3591. IEEE (2016)

[3] https://www.kaggle.com/retailrocket/ecommerce-dataset

[3] Bader-El-Den, M., Teitei, E., Perry, T.: Biased random forest for dealing with the class imbalance problem. IEEE transactions on neural networks and learning systems **30**(7), 2163–2172 (2018)

[4] Bandyopadhyay, S., Thakur, S.: Product prediction and recommendation in e-commerce using collaborative filtering and artificial neural networks: A hybrid approach. In: Intelligent Computing Paradigm: Recent Trends, pp. 59–67. Springer (2020)

[5] Barkan, O., Koenigstein, N.: Item2vec: neural item embedding for collaborative filtering. In: 2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP). pp. 1–6. IEEE (2016)

[6] Bogina, V., Kuflik, T., Mokryn, O.: Learning item temporal dynamics for predicting buying sessions. In: Proceedings of the 21st International Conference on Intelligent User Interfaces. pp. 251–255. ACM (2016)

[7] Cerisara, C., Kral, P., Lenc, L.: On the effects of using word2vec representations in neural networks for dialogue act recognition. Computer Speech & Language **47**, 175–193 (2018)

[8] Chen, C., Hou, C., Xiao, J., Wen, Y., Yuan, X.: Enhancing purchase behavior prediction with temporally popular items. IEICE TRANSACTIONS on Information and Systems **100**(9), 2237–2240 (2017)

[9] Chen, C., Hou, C., Xiao, J., Yuan, X.: Purchase behavior prediction in e-commerce with factorization machines. IEICE TRANSACTIONS on Information and Systems **99**(1), 270–274 (2016)

[10] Chu, Y., Yang, H.K., Peng, W.C.: Predicting online user purchase behavior based on browsing history. In: 2019 IEEE 35th International Conference on Data Engineering Workshops (ICDEW). pp. 185–192. IEEE (2019)

[11] Collell, G., Prelec, D., Patil, K.R.: A simple plug-in bagging ensemble based on threshold-moving for classifying binary and multiclass imbalanced data. Neurocomputing **275**, 330–340 (2018)

[12] Dacrema, M.F., Cremonesi, P., Jannach, D.: Are we really making much progress? a worrying analysis of recent neural recommendation approaches. In: Proceedings of the 13th ACM Conference on Recommender Systems. pp. 101–109 (2019)

[13] Esmeli, R., Bader-El-Den, M., Abdullahi, H.: Improving session based recommendation by diversity awareness. In: UK Workshop on Computational Intelligence. pp. 319–330. Springer (2019)

[14] Esmeli, R., Bader-El-Den, M., Mohasseb, A.: Context and short term user intention aware hybrid session based recommendation system. In: 2019 IEEE International Symposium on INnovations in Intelligent SysTems and Applications (INISTA). pp. 1–6. IEEE (2019)

[15] Grbovic, M., Radosavljevic, V., Djuric, N., Bhamidipati, N., Savla, J., Bhagwan, V., Sharp, D.: E-commerce in your inbox: Product recommendations at scale. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1809–1818. ACM (2015)

[16] Hernandez, J., Carrasco-Ochoa, J.A., Martínez-Trinidad, J.F.: An empirical study of oversampling and undersampling for instance selection methods on imbalance datasets. In: Iberoamerican Congress on Pattern Recognition. pp. 262–269. Springer (2013)

[17] Hidasi, B., Karatzoglou, A., Baltrunas, L., Tikk, D.: Session-based recommendations with recurrent neural networks. arXiv preprint arXiv:1511.06939 (2015)

[18] Jannach, D., Ludewig, M.: When recurrent neural networks meet the neighborhood for session-based recommendation. In: Proceedings of the Eleventh ACM Conference on Recommender Systems. pp. 306–310 (2017)

[19] Jannach, D., Ludewig, M., Lerche, L.: Session-based item recommendation in e-commerce: on short-term intents, reminders, trends and discounts. User Modeling and User-Adapted Interaction **27**(3-5), 351–392 (2017)

[20] Katarya, R., Verma, O.P.: An effective collaborative movie recommender system with cuckoo search. Egyptian Informatics Journal **18**(2), 105–112 (2017)

[21] Krawczyk, B., Jeleń, Ł., Krzyżak, A., Fevens, T.: Oversampling methods for classification of imbalanced breast cancer malignancy data. In: International Conference on Computer Vision and Graphics. pp. 483–490. Springer (2012)

[22] Krismayer, T., Schedl, M., Knees, P., Rabiser, R.: Predicting user demographics from music listening information. Multimedia Tools and Applications **78**(3), 2897–2920 (2019)

[23] Liu, Q., Zeng, Y., Mokhosi, R., Zhang, H.: Stamp: short-term attention/memory priority model for session-based recommendation. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 1831–1839 (2018)

[24] Lo, C., Frankowski, D., Leskovec, J.: Understanding behaviors that lead to purchasing: A case study of pinterest. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. pp. 531–540. ACM (2016)

[25] Ludewig, M., Mauro, N., Latifi, S., Jannach, D.: Performance comparison of neural and non-neural approaches to session-based recommendation. In: Proceedings of the 13th ACM Conference on Recommender Systems. pp. 462–466 (2019)

[26] Martínez, A., Schmuck, C., Pereverzyev Jr, S., Pirker, C., Haltmeier, M.: A machine learning framework for customer purchase prediction in the non-contractual setting. European Journal of Operational Research **281**(3), 588–596 (2020)

[27] Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp. 3111–3119 (2013)

[28] Mohasseb, A., Bader-El-Den, M., Liu, H., Cocea, M.: Domain specific syntax based approach for text classification in machine learning context. In: 2017 international conference on machine learning and cybernetics (ICMLC). vol. 2, pp. 658–663. IEEE (2017)

[29] Mokryn, O., Bogina, V., Kuflik, T.: Will this session end with a purchase? inferring current purchase intent of anonymous visitors. Electronic Commerce Research and Applications **34**, 100836 (2019)

[30] Musto, C., Semeraro, G., de Gemmis, M., Lops, P.: Learning word embeddings from wikipedia for content-based recommender systems. In: European Conference on Information Retrieval. pp. 729–734. Springer (2016)

[31] Ozsoy, M.G.: From word embeddings to item recommendation. arXiv preprint arXiv:1601.01356 (2016)

[32] Parkhimenka, U., Tatur, M., Zhvakina, A.: Heuristic approach to online purchase prediction based on internet store visitors classification using data mining methods. In: 2017 International Conference on Information and Digital Technologies (IDT). pp. 304–307. IEEE (2017)

[33] Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)

[34] Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). pp. 2227–2237. Association for Computational Linguistics, New Orleans, Louisiana (Jun 2018). https://doi.org/10.18653/v1/N18-1202, https://www.aclweb.org/anthology/N18-1202

[35] Qiu, J., Lin, Z., Li, Y.: Predicting customer purchase behavior in the e-commerce context. Electronic commerce research **15**(4), 427–452 (2015)

[36] Rong, X.: word2vec parameter learning explained. arXiv preprint arXiv:1411.2738 (2014)

[37] Salehinejad, H., Rahnamayan, S.: Customer shopping pattern prediction: A recurrent neural network approach. In: 2016 IEEE Symposium Series on Computational Intelligence (SSCI). pp. 1–6. IEEE (2016)

[38] Sheil, H., Rana, O., Reilly, R.: Predicting purchasing intent: automatic feature learning using recurrent neural networks. arXiv preprint arXiv:1807.08207 (2018)

[39] Terán, L., Mensah, A.O., Estorelli, A.: A literature review for recommender systems techniques used in microblogs. Expert Systems with Applications **103**, 63–73 (2018)

[40] Wang, D., Liang, Y., Xu, D., Feng, X., Guan, R.: A content-based recommender system for computer science publications. Knowledge-Based Systems **157**, 1–9 (2018)

[41] Xue, B., Fu, C., Shaobin, Z.: A study on sentiment computing and classification of sina weibo with word2vec. In: 2014 IEEE International Congress on Big Data. pp. 358–363. IEEE (2014)

[42] Yadav, S., Nagpal, S., et al.: An improved collaborative filtering based recommender system using bat algorithm. Procedia computer science **132**, 1795–1803 (2018)

[43] Zhang, D., Xu, H., Su, Z., Xu, Y.: Chinese comments sentiment classification based on word2vec and svmperf. Expert Systems with Applications **42**(4), 1857–1863 (2015)

[44] Zheng, B., Liu, B.: A scalable purchase intention prediction system using extreme gradient boosting machines with browsing content entropy. In: 2018 IEEE International Conference on Consumer Electronics (ICCE). pp. 1–4. IEEE (2018)