01 Jan 2021

Identifying Consumer Preferences From User-generated Content On Amazon.com By Leveraging Machine Learning

Jikhan Jeong
Missouri University of Science and Technology, jikhan.jeong@mst.edu

Recommended Citation

Jeong, J. (2021). Identifying Consumer Preferences From User-generated Content On Amazon.com By Leveraging Machine Learning. *IEEE Access, 9*, pp. 147357-147396. Institute of Electrical and Electronics Engineers.
The definitive version is available at https://doi.org/10.1109/ACCESS.2021.3123301

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 16, 2021, accepted October 21, 2021, date of publication October 27, 2021, date of current version November 8, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3123301

Identifying Consumer Preferences From User-Generated Content on Amazon.Com by Leveraging Machine Learning

JIKHAN JEONG
Department of Data Science and Business Analytics, Florida Polytechnic University, Lakeland, FL 33805, USA
e-mail: jikhan.jeong@wsu.edu

ABSTRACT Inexperienced consumers may have high uncertainty about experience goods that require technical knowledge and skills to operate effectively; therefore, experienced consumers' prior reviews can be useful for inexperienced consumers. However, one-sided review systems (e.g., Amazon) only provide the opportunity for consumers to write a review as a buyer and contain no feedback from the seller's side, so the information displayed about individual buyers is limited. Therefore, this study analyzes consumers' digital footprints (DFs) for programmable thermostats to identify and predict unobserved consumer preferences, using a dataset of 141 million Amazon reviews. This paper proposes novel approaches (1) to identify unobserved consumer characteristics and preferences by analyzing the target consumers' and other prior reviewers' DFs; (2) to extract product-specific product content dimensions (PCDs) from review text data; (3) to predict individual consumers' sentiment before they make a purchase or write a review; (4) to classify consumers' sentiment toward a specific PCD by using context-based word embedding and deep learning models. Overall, this approach developed in this paper is applicable, scalable, and interpretable for distinguishing important drivers of consumer reviews for different goods in a specific industry and can be used by industry to design customer-oriented marketing strategies.

INDEX TERMS Online product review, consumer behavior, natural language prediction, machine learning.

I. INTRODUCTION

In recent years, big data analysis has experienced remarkable growth. This growth has been fostered by innovations in computation performance and remarkable successes with artificial intelligence (AI) algorithms. Additionally, these advances have benefitted from increasing volume, diversity, and value of the data.

There are two types of big data: structured data (which have a well-defined data type) and unstructured data (which lack a well-defined data type, such as image, voice, video, and text). Online product reviews generated by consumers contain both structured and unstructured data. For example, while consumers' product star ratings fall into the category of structured data, their written reviews are unstructured data. User-generated online product review data are free, easy to access, and can provide useful information for inexperienced consumers because they contain feedback from actual consumers who reveal their preferences for products. Such data are quite different from the feedback provided by focus groups or experts.

When a consumer purchases a product through the online retail market, there is uncertainty about the quality of product because the consumer is not in physical contact with it. By leveraging the information from prior review data, inexperienced consumers can reduce their search cost and uncertainty about product quality. Firms can also employ user-generated review content to estimate individual consumer preferences, needs, satisfaction, and complaints and to design, develop, and promote new products. For example, Timoshenko and Hauser [1] demonstrated how to identify consumer needs from user-generated review text on Amazon.

Liu *et al.* [2] suggest that review data are more likely to be influential for consumers when the product group has more competition, a shorter product history, and weaker brand power. Accordingly, inexperienced consumers may have high uncertainty about experience goods when new innovative firms enter the market. Consumers' uncertainty about product

The associate editor coordinating the review of this manuscript and approving it for publication was Liangxiu Han.

quality is relatively higher when they purchase an experience good than a search good, because they may not know the product quality before they make a purchase.

This study investigates Amazon's reviews for a specific experience good (programmable thermostats) requiring enough technical knowledge and skills to install, set up, program, and use it. Products that require technical knowledge and skills to operate (e.g., thermostats) can be difficult for consumers to evaluate before purchase or even after adequately installing the product. Thermostats require time for purchasing consumers to assess the suitability for their needs because people usually do not know their real-time energy consumption, the cost, and the amount of energy saving that a new thermostat can provide in the early stages of thermostat usage. This means that thermostat consumers typically face high uncertainty, and ease of usage and consumer support services are essential for inexperienced consumers to mitigate their concerns and difficulties.

In addition, programmable thermostats (PTs) are not frequently purchased, and malfunctioning could cause flaws in other connected devices, additional repair costs, and physical discomfort. Further, the frequency of and exposure to thermostat advertisements are relatively lower than in other popular research subject products (e.g., movies, music, and books), so the sources of information on thermostats' product quality are less diverse than those on books, music, and movies.

Consumer uncertainty may be higher than normal when disruptive innovation happens because innovative new firms (e.g., Nest) enter the market, introduce innovative products (e.g., Wi-Fi thermostats that can provide remote access and control), and compete with the incumbent firms (e.g., Honeywell). Nest entered the market by releasing the first generation of its learning smart-thermostat on October 25, 2011, and it has been available to purchase from Amazon since December 15, 2011. The Nest released the second generation on October 2, 2012, and it was available from Amazon on the day of release. The Nest's first learning thermostat is an example of disruptive innovation and the internet of things (IoTs) for smart homes [3], [4]. In this regard, inexperienced consumers may have high uncertainty not only due to the required technological knowledge and skills but also to changes in the market structure and competition. This combination of these factors makes thermostats an ideal subject for studying the utility of online reviews to consumers; thermostats can be technically challenging, are of high importance to a home, and potential buyers have few avenues for gaining experience or information prior to purchase.

There are a number of ways to include preferences in models of consumer choice. Revealed-preference methods reflect the actual consumer choices in a real-life situation, while stated-preference methods reflect respondents' hypothetical choices in a well-designed survey or field experiment [5]. Prior studies have widely applied both revealed- and stated-preference methods to estimate consumer preferences. However, these methods may not be applicable for studying online reviews' effects on consumer preferences for technical products such as thermostats.

One-sided review systems like used by Amazon, only provide buyers with the opportunity to write a review which buyers can write without any fee [6], [7]. However, the information displayed about reviewers is limited. Consequently, the conventional revealed- and stated-preference methods cannot be used to directly identify unobserved consumer characteristics and preferences from reviews.

This study identifies unobserved consumer characteristics and preferences by extracting: (1) users' and prior other reviewers' digital footprints (DFs) from user-generated content (UGC) and (2) consumers' sentiment toward product content dimensions (PCDs) from review text data. This study defines this approach as the user-generated-preference (UGP) method.

Consumer review and product-specific review data (142.8 million reviews) from He and McAuley [8], gathered between May 1996 and July 2014, are used to generate DFs. In addition, this study identifies consumers' sentiment toward product content dimensions (PCDs) extracted from review text by applying topic modeling and domain expert annotations, while excluding questionable reviews (posted by "suspicious one-time reviewers" and "always-the-same rating reviewers").

After the data preprocessing is discussed, the following three questions are investigated:

1. Can consumers' preferences be identified through the analysis of digital footprints?
2. Can consumers' sentiment be predicted before they make a purchase or write a review?
3. Can consumers' sentiment toward a specific PCD in the review text be classified?

This paper obtains three main results: first, the author finds that the factors that affect consumer ratings are: (a) users' DFs (e.g., average rating across all categories), (b) reviewers' attitudes toward eight product content dimensions (smart connectivity, easiness, energy saving, functionality, support, price value, privacy, and the Amazon's service quality effect), and (c) other prior reviewers DFs (e.g., length of the review summary). Second, extreme gradient boosting (XGBoost) is found to obtain the highest performance for predicting the sentiment of potential consumers before they make a purchase or write a review. Third, a convolutional neural network (CNN) on top of Bidirectional Encoder Representations from Transformers (BERT) embedding shows the highest performance for classifying consumers' sentiment toward a specific PCD.

These findings will potentially be helpful for firms to identify consumer preferences, predict potential consumer sentiment, extract product content dimensions for a specific product group from review text, and classify consumers' sentiment toward a specific product content dimension. Firms often want to know potential individual consumers' preferences concerning target product groups in a specific industry (e.g., thermostats) instead of a general product category

TABLE 1. Previous literature.

Source	Data	Target	Method*	Related findings
Anderson and Magruder (2012) [9]	Yelp	Restaurant	RDD	1. If the average rating increases, the frequency of consumer flows will increase 2. The effect of ratings is high when consumers have fewer alternative information sources
Chen (2018) [10]	Yelp, Medicare	Physician	DiD LDA	1. Increasing the average ratings for a physician increases the physician's revenues and patient volume
Chevalier and Mayzlin (2006) [11]	Amazon Barnes and Noble	Book	DiD	1. A higher rating of reviews may increase relative sales. 2. The impact of a one-star rating on relative sales is greater than that of a five-star rating 3. The statistical significance of the review length variable indicates that consumers read the text in the reviews
Cui, Lui, and Guo (2012) [7]	Amazon	Video game Electronics	Panel model	1. The volume of reviews is more important for the sales of new experience goods than those of search goods. 2. The impact of the volume of reviews decreases over time.
Hu, Liu, and Zhang (2008) [12]	Amazon	Book, DVD Video	Regression	1. The impact of reviews on sales is larger when (a) the reviewer has a better reputation (b) the items were less reviewed by prior reviewers 2. The impact of reviews decreases as the item ages
Liu, Lee, and Srinivasan (2019) [2]	Online retailer in the UK	Home and garden, technology	CNN RDD LDA	1. The effect of review content on sales is high when the average rating increases, the variance of the rating decreases, and the market is more competitive
Luca (2016) [13]	Yelp, WA department of revenue	Restaurant	RDD	1. If the average rating increases, the revenue will increase
Mayzlin, Dover, and Chevalier (2014) [14]	Expedia TripAdvisor	Hotel	Panel model	1. Hotels may have different levels of incentive to write promotional reviews based on their competition and ownership condition
Reimers and Waldfogel (2020) [15]	New York Times, Amazon	Book	Panel model Nested logit	1. Professional critics' and crowd' star ratings affect sales and consumer surplus
Susan and David (2010) [20]	Amazon	CD, MP3, video game	Tobit model	1. Five- or one-star rating reviews are less helpful for experience goods than mild rating reviews
Luca and Zervas (2016) [17]	Yelp	Restaurant	Regression	1. A restaurant that has a weak reputation is more likely to write negative fake reviews of competitors 2. Fraud involving negative fake reviews may increase when the market becomes more competitive
Zhao *et al.* (2013) [18]	US companies	Book	Bayesian model	1. Consumers learn product quality from product reviews compared with their own experience with similar products 2. Fake reviews enhance the uncertainty of consumers
Timoshenko and Hauser (2019) [1]	Amazon	Oral care	CNN W2V	1. Deep learning methods increase the performance of identifying consumer needs from user-generated review sentences
Hu, Pavlou, and Zhang (2006) [21]	Amazon	Book, DVD, Video	Theory	1. Average ratings from reviewers may mislead consumers regarding the quality of the products because ratings often follow a bimodal distribution

* Notes: RDD: regression discontinuity design; DiD: difference in difference; CNN: convolutional neural network; W2V: word2vec; LDA: latent dirichlet allocation).

level (e.g., book). Better short-term predictions of potential consumers' preferences for industry-specific product groups may also help firms to improve their business decisions.

Section 2 describes the prior literature. Section 3 presents the data-preprocessing for cleaning noisy reviews and extracting target reviewers' sentiment toward the product content dimensions. Section 4 describes the discrete choice analysis. Section 5 demonstrates the ex-ante prediction of potential consumers' sentiment. Section 6 shows the sentiment classification of a specific product content dimension. Finally, section 7 offers conclusions.

II. LITERATURE REVIEW

Many previous studies have focused on the impact of reviews on sales [2], [7], [9]–[15]. Most studies have used summary statistics of aggregated review data at the product level (e.g., the average rating for a product, the volume of reviews for a product, and the average review length for a product).

On an individual level, Liu *et al.* [2] extracted product content dimensions from individual review text by using topic modeling. The authors demonstrated the classification of each product content dimension by using deep learning and measured the effect of each product content dimension on sales. Further, Timoshenko and Hauser [1] identified consumer needs from individual review text by using deep learning.

One possible challenge of using online review data is potential noise, bias, or promotional reviews [16]. As shown in Table 1, some previous studies have investigated the impact of ownership, reputation, and market competition on firms' incentives to write a promotional review by analyzing aggregated product level summary data [14], [17].

In contrast to previous research, which has used Amazon's online reviews for general experience goods (e.g., books, DVDs, and music), this study investigates Amazon's online reviews for a specific experience good (programmable thermostats).

To the best of the author's knowledge, there is little current research that addresses how to: (1) identify potential suspicious one-time or always-the-same rating reviewers; (2) estimate unobserved individual reviewers' characteristics from user DFs; (3) evaluate the effect of prior other reviewers' DFs on the target reviewers' ratings; (4) extract latent product content dimensions from review text; (5) predict potential consumers' sentiment before they make a purchase or write a review; and (6) classify reviewers' sentiment toward a product content dimension in the review.

III. DATA PRE-PROCESSING

This study aims to estimate and consumer preferences for the group of Amazon users who write a review by using the review data written by this group while excluding biased reviews (Appendix). Therefore, this paper implements specific data-preprocesses (Appendix) as follow:

Step. 1: Selecting reviews with no missing values,
Step. 2: Cleaning "suspicious one-time reviewers" and "always-the-same-rating reviewers";
Step. 3: Deleting reviewers and reviews for products with no digital footprint (DFs);
Step. 4: Selecting the top 6 from 26 brands;
Step. 5: Identifying five product content dimensions (PCDs) in the review text using LDA; and
Step. 6: Modifying the PCDs by leveraging a domain expert's knowledge.

Zhao *et al.* [18] indicated that fake reviews increase consumers' uncertainty about products and that more believable online reviews of experience goods have a larger effect on consumer choice. Some firms may write positive reviews about their products and negative ones about their rivals' products [14], [17], [19]. Accordingly, deleting potential suspicious reviews during pre-processing is essential to improve the credibility of reviews and reduce consumer uncertainty.

Mayzlin *et al.* [14] defined the "suspicious reviewer" as one who writes a review for a hotel for the first time only during the sample period (October 2011) and showed that their rating distribution is more polarized than that of the entire sample. This study takes suspicious reviewers into account by accessing individual reviewers' prior reviews in different categories over the entire sample period. A "suspicious one-time reviewer" is defined as one who writes only a review for a programmable thermostat (PT) as a first review and does not write reviews for any other products over the entire sample period.

Some reviewers always give a star rating at the same level for all reviewed products in all categories, so their reviews may contain self-selection bias. However, it is possible that the reviewers give the same rating level because the number of reviews is simply small. In this study, an "always-the-same-rating reviewers (ASR)" is a reviewer who writes more than 8 reviews with the same rating level. Only 69 reviewers write more than 8 reviews at the same star rating level (5 stars), and these reviewers' 69 reviews for PTs are removed.

The purpose of this study is to identify latent consumers' characteristics and preferences by analyzing DFs, so the sample group disregards reviewers and programmable thermostats containing no prior DFs. DFs from earlier reviewers (crowd) may have the greatest effect on subsequent reviewers when the reviewer posts his or her first review. This study therefore focuses on the target reviewers' first review of a programmable thermostat. After only selecting the first review of each reviewer for the thermostat group, the total number of reviewers and their first-time reviews is 5,307, and the total number of reviews for all products (including programmable thermostats) written by these reviewers in all categories over the entire sample period is 169,809.

In contrast to previous studies using aggregated review summary statistics at the product level, this study extracts individual reviewers' digital footprints for a specific product group from a dataset of 141 million Amazon reviews. In detail, digital footprints of individual target reviewers and other prior reviewers (the crowd) are extracted from all the reviews in all categories over the entire sample period and this information is used to identify and predict latent consumer preferences and sentiment.

The review text often contains information that is useful for identifying the latent PCDs [2], each reviewer's sentiment, and the direct or indirect reasons for the star rating given. Latent Dirichlet allocation (LDA) [22] is an unsupervised learning model used to identify latent topics and the distribution of these topics in each review. Therefore, the author determines five PCDs in the review text by applying LDA.

Passonneau *et al.* [23] suggested that annotation by experts transfers domain knowledge to machines for better prediction performance. Accordingly, the author (the domain expert) manually annotates 47,763 labeling tasks for the reviewers' sentiment toward each product content dimension (PCD) to transfer domain knowledge to the models into nine PCDs based on domain knowledge and the purpose of the research design (Appendix).

The nine dimensions are: (1) smart-connectivity, (2) easiness, (3) energy saving, (4) functionality, (5) support, (6) perceived price value, (7) privacy, (8) the Amazon effect, and (9) environmental friendliness. The domain expert annotates each reviewer's sentiment toward each PCD to transfer domain knowledge from the expert to the empirical models.

IV. ECONOMETRIC ANALYSIS

Amazon uses five-star ratings from one to five. Reviewers' observable ratings indicate the range of their unobservable

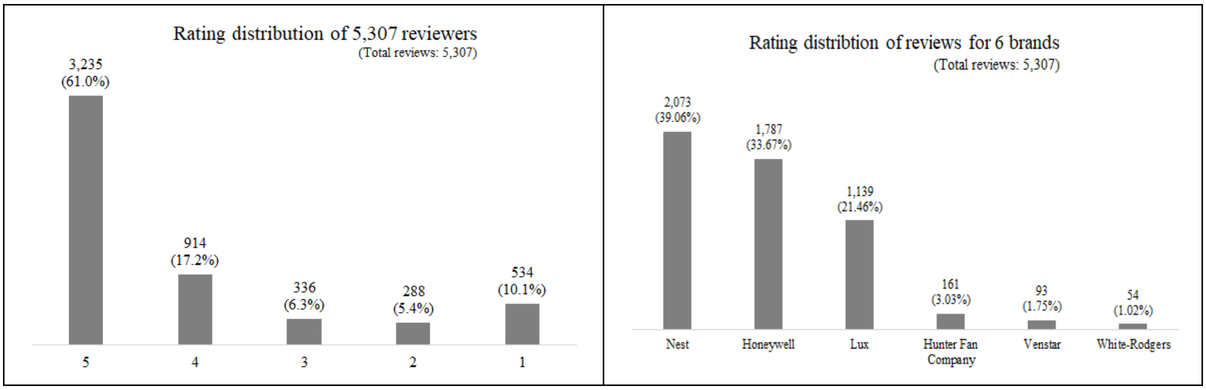


FIGURE 1. Rating distributions of reviews for six major brands.

TABLE 2. Previous literature.

Topic dimensions	Interpretation
Smart-connectivity	The review describes WiFi, including wireless connection issues with software (e.g., App) and hardware (e.g., Heating, ventilation, and air conditioning).
Easiness	The review mentions ease of use, including simplicity of installation, programming, and use.
Energy Saving	The review talks about energy savings, including money savings by reducing energy consumption.
Functionality	The review contains content related to the quality of the setting, controlling, and information related to temperature, time, scheduling, heating, and other devices.
Support	The review focuses on consumer support services before, during, and after they make a purchase.
Price value	The reviews discuss a reviewer's subjective evaluation about the price level compared with the quality, future benefits, and other factors.
Privacy	The reviews indicate privacy concerns related to thermostats.
Amazon Effect	The reviews mention Amazon's service quality, such as Amazon's delivery, consumer support, and refund and replacement policy.
Environmental friendliness	The reviews point out the issues related to carbon emissions and climate change.

continuous preference [24] as follows:

$$\begin{aligned}
R_{ipt} &= 1, \quad \text{if } -\infty < U^*_{ipt} \leq c_1\\
R_{ipt} &= 2, \quad \text{if } c_1 < U^*_{ipt} \leq c_2,\\
R_{ipt} &= 3, \quad \text{if } c_2 < U^*_{ipt} \leq c_3,\\
R_{ipt} &= 4, \quad \text{if } c_3 < U^*_{ipt} \leq c_4,\\
R_{ipt} &= 5, \quad \text{if } c_4 < U^*_{ipt} < \infty.
\end{aligned}$$

The ordered dependent variable, $R_{ipt} \in [1, 5]$, is reviewer i's first star rating for a PT on day t. U^*_{ipt} denotes the unobservable continuous utility of reviewer i for product p on day t. The unknown cutting points (thresholds) are denoted as c_k with the assumption that $c_1 < c_2 < c_3 < c_4$. U^*_{ipt} can be represented as follows:

$$U^*_{ipt} = x'_{ipt}\beta + \rho\varepsilon_{it}, \quad \varepsilon_{it} \sim \text{i.i.d Normal } (0, 1)$$

where x_{it} indicates a vector of independent variables, $\rho > 0$ is a scale function to adjust the variance, and ε_{it} is a homoskedastic error term following a standard normal distribution [25], [26]. Hu *et al.* [21] showed that the star rating distribution of some experience goods (books, DVDs, and videos) follows a bi-modal distribution on Amazon. The frequency of observed star ratings (from 1 to 5 stars) in this study follows a bi-modal distribution, that is a non-normal distribution. However, the cutting points adjust each rating probability (following a normal distribution) to match the observed rating distribution [27].

The ordered probit (OP) model assumes that $\rho = 1$, so there is no scaling effect on the underlying preferences. Some researchers have studied or applied heteroskedasticity to ordered response models [25], [28]–[33].

In contrast to linear regression models, the existence of latent heteroskedasticity will cause inconsistency in the maximum likelihood estimators of OP models [27]. The heteroskedasticity ordered probit (HETOP) model assumes its scaling function to be $\rho_i = \exp(Z'_{it}\gamma)$, where Z_i denotes the regressors for the scaling function and γ are unknown coefficients for Z_{it}. In addition, the variables in x_{it} can overlap with those in Z_{it}; therefore, x^a_{it} denotes the variables involved in both x_{it} and Z_{it} while x^b_{it} denotes the variables that only belong to x_{it}. Unknown parameters are estimated through the maximum likelihood estimation (Appendix).

This study assumes that the reviewers' different prior review experiences and patterns reflect their unobserved characteristics and preferences. The variables are divided into "at time" variables extracted from DFs at t_i; "user DF" variables extract reviewer i's prior reviews across all categories by t_i^b or at t_i^b; and "crowd DF" variables extract the reviews written by other prior reviewers on the PT by $t_{j\neq i}^b$ or at $t_{j\neq i}^b$. The number of prior reviews written by i in each subcategory by t_i^b is denoted as "sum_+ subcategory name" and 32 subcategories are defined by merging similar subcategories during the pre-processing. The category diversity is the Shannon index, for which higher values mean that reviewer i writes reviews in subcategories with greater diversity by t_i^b (Appendix). The digital footprints (DFs) and sentiment variables in this study are defined in Table 3.

As can be seen in Table 4, each model in this section contains a different combination of variables to identify the effects of DFs, sentiments, prices, and the volume of prior reviews on the consumers' star ratings. In particular, the review text data are divided into "review summary (headline)" and "review body". "Review" in this study denotes both the review summary and the review body text. In addition, other ex post reviewers' helpfulness votes for reviewer i's review after t_i are an ex post variable that does not affect the reviewers' star rating at t_i; therefore, this study disregards helpfulness votes for reviews after t_i.

Omitted variables and the existence of heteroskedasticity may cause inconsistency of parameters in OP models [27]. The models in this section contain the variables extracted from DFs and review text data to reduce the omitted variable problem.

The misspecification of the variation function in HETOP models leads to biased parameters [30]. The author compares the empirical results between the HETOP and the OP models with different sets of regressors to check the variation function's misspecification in the HETOP models. The notation "model_o" indicates an OP model and "model_h" indicates a HETOP model. Model_o1 is the base model, which contains only observable variables at t_i.

The Akaike information criterion (AIC) and Bayesian information criterion (BIC) designate the model with fewer parameters and smaller sample sizes as a better-fitted model [27]. A smaller AIC or BIC value means a better model fit. All the HETOP models show better model fits than the OP models with the same set of regressors (Appendix). All the HETOP models also show the existence of heteroskedasticity in the likelihood ratio test.

Surprisingly, the models with price (at the time of web scraping) variables show a lower model fit than the models without price variables. Product prices on Amazon frequently change due to promotions, memberships, and other factors, so the actual price of reviewed products may often differ from the price at the time of web scraping. Further, the actual price at the time of purchasing could be different from the price at the time of writing a review. This price gap between the actual price and the price at the time of web scraping might be a source of inherent bias in the price variables. This study uses the reviewers' sentiment toward the perceived price value dimension as a sentiment variable.

In detail, the sign of coefficients for variables in OP models reflect the sign of the marginal effect with the extreme star ratings ($R_{ipt} = 5$ and $R_{ipt} = 1$). In the HETOP models, the sign of the coefficients for x_{it}^a variables (that involved in both x_{it} and Z_{it}) reflect the sign of the marginal effects for the x_{it}^a variables with the extreme ratings. However, the sign of the coefficients for x_{it}^b variables (that only belong to x_{it}) does not directly reflect the sign of the marginal effects with any star ratings. In this study, all the variables in the HETOP models are x_{it}^a variables, excluding six x_{it}^b variables consisting of the reviewer's average star rating by t_i^b and five brand dummies.

The interpretations for the most satisfied consumers (five-star reviewers) are based on statistically significant variables in model_h2 (the main model for interpretation) and model_h4 (the model for interpretation of the volume of prior reviews in each subcategory).

Based on the user DF variables in model_h2, the probability that a reviewer will give a five-star rating to the reviewed PT will decrease if the reviewer writes a longer review summary or body and has a greater volume of prior reviews in all categories.

In contrast, the probability of a reviewer giving a five-star rating will increase if the reviewer has a higher variance of review summary length in prior reviews. In addition, the reviewer's average star rating in prior reviews has a positive influence on the probability of the reviewer giving a five-star rating. Even though the direct economic interpretation is limited, the coefficient of the reviewer's average star rating is the largest among the statistically significant variables in model_h2.

With other prior reviewers' DF variables in model_h2, the probability of a reviewer giving a five-star rating for a PT increases with increased variability in length of prior review summaries for the PT.

In contrast, the probability that a reviewer will give a five-star rating for a PT decreases as the average length of the prior review summary increases. Chevalier and Mayzlin [11] suggested that the statistical significance of the review length variable indicates that consumers read the text in the reviews. Here, this point suggests that a reviewer who gives the extreme ratings (a 1-star or 5-star rating) may respond to prior reviewers' review summary.

Based on the reviewers' sentiment toward product content dimensions (PCDs) extracted from the review text, the probability of a reviewer giving a five star-rating increases if the reviewer has a positive attitude toward "smart connectivity," "easiness," "energy saving," "functionality," "support," "price value," "privacy," and "Amazon effect" dimensions. The results of the sentiment variables indicate that consumers prefer "smarter" and "easier-to-use" PTs. In addition, these consumers prefer PTs made by firms that provide better support for consumers. Therefore, firms need to consider not

TABLE 3. Variables generated from user and crowd DFs (N= 5,307).

Variable	Description
rating (dependent)	i (the reviewer)'s five-scale star rating for a PT at t_i*
sum_len	i's length of review summary (headline) at t_i
rev_len	i's length of review body at t_i
title_len	The length of tittle for the PT reviewed by i at t_i
desc_len	The length of description for the PT reviewed by i at t_i
Nest	Brand dummy for the Nest (base group is White Roger)
Honey	Brand dummy for the Honeywell
Hunter	Brand dummy for the Hunter Fan
Lux	Brand dummy for the Lux
Venstar	Brand dummy for the Venstar
Price	p (the PT reviewed by i at t_i)'s price (at the time of web scrapping)
u_avg_p_dfs	i's average price for reviewed products in all categories by t_i^b*
u_sd_p_dfs	i's SD of price for reviewed products in all categories by t_i^b
u_max_p_dfs	i's the highest price among reviewed products in all categories by t_i^b
u_help_dfs	The number of helpfulness upvote for i in all categories by t_i^b
u_no_help_dfs	The number of helpfulness downvote for i in all categories by t_i^b
u_avg_len_sum	i's average length of summary in all categories by t_i^b
u_sd_len_sum	i's SD of length of summary in all categories by t_i^b
u_avg_len_rev	i's average length of review body in all categories by t_i^b
u_sd_len_rev	i's SD of length of review body in all categories by t_i^b
sum_sub-category	i's number of reviews in sub-category by t_i^b where thirty-two sub-categories are: {the amazon instant video; the appliance; the apps for android category; the art crafts; the automotive; the baby; the beauty; the book; the kindle; the cds and vinyl; the cell phones; the clothes, shoes, jewelry; the computer; the digital music; the electronics; the gift cards; the grocery gourmet food; the health personal care; the home kitchen; the industry specific; the kindle store; the magazine subscription; the move and tv; the musical instrument; the office products; the patio, lawn, and garden; the pet supplies; the software; the spots and outdoors; the tools & home; the tops and games; the video games}
u_cum_reviews	i's number of reviews in all categories by t_i^b
u_cate_diversity	Shanon index for i's category diversity of reviews posted by t_i^b
u_avg_rating	i's average star rating in all categories by t_i^b
u_sd_rating	i's SD of star rating in all categories by t_i^b
c_cum_reviews	p's number of crowd's reviews by t_i^b
c_avg_rating	p's average rating of crowd by $t_{j\neq i}^b$*
c_sd_rating	p's SD of crowd's rating by $t_{j\neq i}^b$
c_avg_len_sum	p's average length of review summary written by crowd until $t_{j\neq i}^b$
c_sd_len_sum	p's SD of review summary written by crowd until $t_{j\neq i}^b$
c_avg_len_rev	p's average length of review body written by crowd until $t_{j\neq i}^b$
c_sd_len_rev	p's SD for the length of review body written by crowd until $t_{j\neq i}^b$
c_rating_rec	p's average rating of crowd at $t_{j\neq i}^b$
c_len_sum_rec	p's the length of review summary written by a crowd at $t_{j\neq i}^b$
c_len_rev_rec	p's the length of review body written by a crowd at $t_{j\neq i}^b$
Day	Day dummies for t_i and base day is Monday (0)
Month	Month dummies for t_i and base month is January (1)
Year	Year dummies for t_i and base year is 2005
Holiday	US holiday dummies and base is not holiday (0)
Interval	The time interval between p's the day of first review and t_i
nest_avail	Dummy for the first day of the Nest's PT on Amazon (Dec 15, 2011)
smart_con	i's sentiment of p's smart connectivity in i's review at t_i
Easy	i's sentiment of p's easiness in i's review at t_i
Save	i's sentiment of p's energy saving in i's review at t_i
Func	i's sentiment of p's functionality in i's review at t_i
Support	i's sentiment of p's support in i's review at t_i
price value	i's sentiment of p's perceived price value in i's review at t_i
privacy	i's sentiment of p's privacy issues in i's review at t_i
Amazon	i's sentiment of p's Amazon effect in i's review at t_i
Env	i's sentiment of p's environmental friendliness in i's review at t_i

* Notes: t_i = the day when reviewer i wrote a review about a PT (p) for the first time; ** t_i^b = the most recent day when reviewer i wrote a review before t_i; $t_{j\neq i}^b$ = the most recent day when the prior other reviewer j wrote a review before t_i; and symbol u in front of the variables (e.g., u_avg_rating) indicates user DFs while c indicates crowd DFs (e.g., c_avg_rating).

TABLE 4. Empirical results from the HETOP and OP models (Appendix).

Variable	model_o1	model_h2	model_h3	model_h4	model_h5
sum_len	-0.010***	-0.008***	-0.007***	-0.008***	-0.008***
rev_len	-0.000***	-0.000***	-0.000***	-0.000***	-0.000***
title_len	0.003	0.003	0.002	0.002	0.002
desc_len	-0.000	-0.000**	-0.000**	-0.000**	-0.000**
Nest	0.463***	0.170	0.033	0.191	0.050
Honey	0.426***	0.431**	0.393*	0.492**	0.460**
Hunter	0.235	-0.008	-0.022	-0.034	-0.050
Lux	0.469***	0.555**	0.531**	0.594**	0.578**
Venstar	0.648***	0.428	0.324	0.488	0.386
Holiday	0.028	0.188	0.187	0.240	0.241
help_dfs		0.004	0.004	0.005	0.005
no_help_dfs		-0.011	-0.011	-0.013	-0.013
u_avg_len_sum		-0.002	-0.002	-0.002	-0.002
u_sd_len_sum		0.009*	0.010*	0.010	0.011*
u_avg_len_rev		-0.000	-0.000	0.000	0.000
u_sd_len_rev		0.000	0.000	0.000	0.000
cum_reviews		-0.001**	-0.001**		
cate_diversity		0.002	-0.016	-0.016	-0.022
u_avg_rating		0.225***	0.223***	0.254***	0.255***
u_sd_rating		0.006	0.009	0.014	0.015
c_avg_rating		0.120	0.102	0.137	0.123
c_sd_rating		-0.140	-0.141	-0.159	-0.160
c_cum_reviews		0.000	-0.000	0.000	-0.000
c_avg_len_sum		-0.022*	-0.020*	-0.028**	-0.027*
c_sd_len_sum		0.030**	0.028**	0.034**	0.032**
c_avg_len_rev		0.000	0.000	0.000	0.000
c_sd_len_rev		0.000	0.000	0.000	0.000
c_rating_rec		0.008	0.008	0.010	0.010
c_len_sum_rec		-0.001	-0.001	-0.001	-0.001
c_len_rev_rec		-0.000	-0.000	-0.000	-0.000
Interval		-0.000	-0.000	-0.000	-0.000
nest_avail		0.152	0.133	0.134	0.125
smart_con		0.699***	0.689***	0.826***	0.824***
Easy		0.806***	0.796***	0.937***	0.937***
Save		0.713***	0.704***	0.858***	0.859***
Func		1.407***	1.390***	1.621***	1.621***
Support		1.147***	1.133***	1.326***	1.328***
price_value		0.675***	0.673***	0.765***	0.769***
Privacy		1.915***	1.889***	2.337***	2.333***
Amazon		0.298*	0.296*	0.331*	0.333*
Env		0.058	0.050	0.170	0.184
Price			0.001		0.001
u_avg_p_dfs			0.000		0.000
u_sd_p_dfs			-0.000		-0.000
u_max_p_dfs			0.000		0.000
sum_amz_video				-0.667*	-0.664*
sum_appliance				0.227*	0.224*
sum_apps				-1.995*	-1.998*
sum_cellphone				-0.051*	-0.049*
sum_clothes				-0.091*	-0.091*
sum_grocery				-0.043**	-0.043**
sum_healthcare				0.055**	0.056**
sum_magazine				-0.367*	-0.367*
sum_pet_supp				-0.053*	-0.052*

TABLE 4. ***(Continued.)*** **Empirical results from the HETOP and OP models (Appendix).**

Z. u_avg_rating		-0.032*	-0.032*	-0.031	-0.030
Z. nest		0.544***	0.529***	0.681***	0.679***
Z. honey		0.398**	0.381**	0.531***	0.528***
Z. lux		0.386**	0.369**	0.491**	0.489**
Z. hunter		0.582***	0.569***	0.726***	0.725***
Z. venstar		0.229	0.200	0.364	0.351
LR tests, $X^2(6)$		28.69***	28.93***	34.37***	34.48***
Loglikelihood	-6,046.585	-4,003.836	-4,002.670	-3,977.845	-3,977.223
AIC	12,171.169	8,159.673	8,165.341	8,169.689	8,176.446
BIC	12,427.656	8,659.494	8,691.468	8,873.385	8,906.448

* Notes: *P-value* = **p* < .1; ***p* < .05; ****p* < .01; statistically insignificant variables represent the volume of prior reviews in each sub category by t_i^b and time dummies are not presented; "Z.variable" indicates a regressor of the variation function; LR test indicates likelihood ratio tests for the existence of heteroskedasticity in the model; The sample size in this section is 5,306 as the number of samples in 2005 is 1.

only developing smarter products but also making them easier for consumers to use with better consumer support programs.

This same group of consumers also consider a PT's energy saving capacity, functionality, and perceived price value. Interestingly, privacy also affects these consumers' preferences, as they may be concerned about the information stored and transmitted by wireless smart thermostats. Firms may need to mitigate consumers' concerns about their privacy with respect to energy consumption and life pattern data.

To the best of the author's knowledge, this is the first study to investigate the effect of online retail market service quality on consumers' sentiments. Amazon's better service quality (such as faster delivery, better consumer service, and flexible refund policy) may increase the probability of a reviewer giving a five-star rating. This service-related result supports the idea that online retail market service quality may influence consumers' preferences as well. Therefore, without considering the effect of online market service quality on the reviewers, the estimation of consumer preferences may lead to upward or downward bias. In contrast with the service dimension, the "environmental friendliness" dimension proved to be statistically insignificant.

Model_h4 contains thirty-two variables for the volume of prior reviews in each subcategory instead of the volume of prior reviews in all categories, like model_h2. The results of model_h4 indicate that the probability of a reviewer giving a five-star rating increases if the reviewer has written a larger volume of prior reviews for products in the "appliance" and "health care and personal care" categories by t_i^b. For example, reviewers who have a high volume of prior reviews for products in the "appliance" category might have more technical knowledge and experience with hardware devices. In addition, thermostats are home energy control devices designed to keep the ideal temperature for consumers' comfort within their homes, so consumers who have a greater volume of prior reviews for products in the "health care and personal care" category may have better knowledge related to thermostats.

In contrast, the probability of a reviewer giving a five-star rating decreases if the reviewer writes a higher volume of reviews for products in the "Amazon instant video," "apps," "cell phones," "clothes," "groceries," "magazine subscriptions," and "pet supplies" categories. While these data-driven interpretations are subjective, they do show how to use DFs to identify latent consumer characteristics.

A. MARGINAL ANALYSIS

Generally, marginal effect analysis is an appropriate way to interpret each parameter in OP models due to non-linearity. Table 5 shows the marginal effect of key variables (model_h2) at the average value of one company's reviewers (Nest, during June 2014).

The sign of the marginal effect of x_{it}^a for the extreme ratings is the same as the sign of the coefficient of those variables in model h2. Accordingly, the average star rating of the reviewers by t_i^b (only one continuous x_{it}^b variable) shows the same sign as the coefficient of this variable for the extreme ratings in model_h2. In contrast, the marginal effect of binary dummy variables for each brand (dummy type of x_{it}^b) shows different signs from the coefficient for these dummies over the star ratings.

In terms of other prior reviewers' (crowd) DF variables, the brand dummy variables show different patterns of marginal effects for each star rating. The marginal effect of the Nest brand dummy shows a negative influence on the probability of a reviewer giving a five-star rating; otherwise, it shows a positive influence on the probability of the reviewer's other star ratings. Increasing the crowd's average length of review summary for the PT will decrease the probability of the reviewer giving a five-star rating. In contrast, increasing the crowd's variance of the review summary length for the PT will increase the probability of a five-star rating.

In terms of the reviewers' sentiment toward the nine PCDs, eight sentiment variables are statistically significant, while the environmental friendliness dimension is not. The sentiment variables show a positive relationship with the probability of a five-star rating; however, the sentiment variables

TABLE 5. Marginal effect of the key variables in model_h2 (Appendix).

	Rating 1	Rating 2	Rating 3	Rating 4	Rating 5
sum_len	0.0000965***	0.0001758***	0.0003727***	0.0011106***	-0.0017556***
	(0.0000324)	(0.0000482)	(0.0000897)	(0.000243)	(0.0003798)
rev_len	0.0000019***	0.0000035***	0.0000075***	0.0000223***	-0.0000353***
	(0.00000065)	(0.00000096)	(0.00000178)	(0.00000477)	(0.00000749)
desc_len	0.00000096**	0.0000017*	0.0000037***	0.000011***	-0.0000174
	(0.00000041)	(0.00000068)	(0.00000137)	(0.00000403)	(0.00000627)
nest	0.0069402***	0.0155956***	0.0359091***	-0.0036256	-0.0548193
	(0.0021822)	(0.0036217)	(0.0104679)	(0.093566)	(0.0988674)
honey	0.0256201	0.0141422	0.0085583	-0.0587262***	0.0104056
	(0.0234841)	(0.0098896)	(0.0095073)	(0.0202738)	(0.042016)
lux	0.0203594	0.0107719	0.0041348	-0.0669442***	0.0316781
	(0.0212434)	(0.0100643)	(0.0104108)	(0.0195704)	(0.0447294)
hunter	0.0782568*	0.0320287***	0.0247976***	-0.0532644	-0.0818186*
	(0.0431248)	(0.0073512)	(0.0080534)	(0.0327548)	(0.0428057)
venstar	0.0076163	0.0043517	-0.0007767	-0.0537036**	0.0425123
	(0.0175559)	(0.0131642)	(0.0157755)	(0.0261312)	(0.056631)
u_sd_len_sum	-0.0001197	-0.0002181*	-0.0004624*	-0.0013777*	0.0021779*
	(0.0000742)	(0.0001274)	(0.0002612)	(0.0007574)	(0.0012023)
cum_review	0.0000123*	0.0000224**	0.0000475**	0.0001416**	-0.0002238**
	(0.00000654)	(0.0000111)	(0.0000225)	(0.000065)	(0.0001029)
u_avg_rating	-0.0044397***	-0.0072649***	-0.0139481***	-0.032701***	0.0583536***
	(0.0014089)	(0.0017163)	(0.0024774)	(0.0051835)	(0.007571)
c_avg_len_sum	0.0002848*	0.000519*	0.0011001*	0.0032782**	-0.0051821**
	(0.00016)	(0.000275)	(0.0005657)	(0.0016594)	(0.0026128)
c_sd_len_sum	-0.0003876**	-0.0007063**	-0.0014971**	-0.004461**	0.0070518**
	(0.0001869)	(0.0003109)	(0.0006273)	(0.0018085)	(0.0028552)
smart_con	-0.0089893***	-0.0163805***	-0.034722***	-0.1034653***	0.1635571***
	(0.002545)	(0.0032834)	(0.0050567)	(0.0104058)	(0.0162277)
easy	-0.0103636***	-0.0188848***	-0.0400303***	-0.1192833***	0.188562***
	(0.0027898)	(0.0035047)	(0.0051561)	(0.0090109)	(0.0134368)
save	-0.0091721***	-0.0167137***	-0.0354282***	-0.1055698***	0.1668838***
	(0.0025736)	(0.0033788)	(0.0054014)	(0.0121838)	(0.0185038)
func	-0.0180985***	-0.0329797***	-0.0699073***	-0.2083114***	0.3292969***
	(0.0047887)	(0.005948)	(0.0085658)	(0.0140342)	(0.01997)
support	-0.01475***	-0.0268779***	-0.0569733***	-0.1697703***	0.2683714***
	(0.0040622)	(0.005184)	(0.0077551)	(0.0138356)	(0.0217078)
price_value	-0.0086866***	-0.015829***	-0.0335529***	-0.0999817***	0.1580502***
	(0.0024035)	(0.0031331)	(0.0049424)	(0.0105698)	(0.0160569)
privacy	-0.0246247***	-0.044872***	-0.0951157***	-0.2834277***	0.4480401***
	(0.0077094)	(0.0110772)	(0.0198971)	(0.0531793)	(0.0820835)
amazon	-0.0038388*	-0.0069952*	-0.0148277*	-0.044184**	0.0698457**
	(0.0022045)	(0.0037641)	(0.0077086)	(0.0224601)	(0.03551)
env	-0.00075	-0.0013668	-0.0028971	-0.008633	0.0136469
	(0.0089839)	(0.0163656)	(0.0346832)	(0.1033051)	(0.1633324)

* Notes: *P-value* = *p < 0.1; **p < 0.05; ***p < 0.01; only consider statistically significant variables or related variables.

have a negative relationship with the other star ratings. If a reviewer has more positive sentiment toward smart connectivity, easiness, energy saving, functionality, support, pricy value, and privacy for programmable thermostats and Amazon's service quality, the probability of writing a five-star rating will increase while other star ratings will decrease.

B. ROBUSTNESS

All the models containing digital footprints (DFs) and sentiment variables show a much better model fit than the base model_o1 (which contains only observable variables at t_i). Nonetheless, latent omitted variable bias is still a concern because a one-sided review system cannot provide actual socio-demographic information about the reviewers.

To account for potential omitted variable bias, the robustness test in this study follows Mayzlin *et al.* approaches [14]. The first step is to compare the coefficients of the key variables between the model without control variables (the base model) and the model with control variables (the control model). If the signs of the coefficients for the key variables are the same and the magnitudes of the coefficients for the key variables are similar between the base and the control model, the effect of omitted variables on the coefficients of the key variables may be relatively small. In this case, the omitted variable problem might be neglectable for estimating the coefficients of the key variables.

As shown in Table 6, the sign of the coefficients for the statistically significant key variables is the same in the control and the base models. The magnitudes of the coefficients for the key variables are also similar in the control and the base models. These empirical results indicate that the omitted variable problem might be lessened by adding digital

TABLE 6. Robustness test for the HETOP models.

Variable	Base (47 variables)	model_h with control (66 variables)
sum_len	-0.008*** (0.002)	-0.008*** (0.002)
rev_len	-0.000*** (0.000)	-0.000*** (0.000)
desc_len	-0.000** (0.000)	-0.000** (0.000)
nest	0.286 (0.199)	0.170 (0.248)
honey	0.425** (0.206)	0.431** (0.213)
hunter	-0.104 (0.237)	-0.008 (0.265)
lux	0.498** (0.220)	0.555** (0.234)
venstar	0.491* (0.271)	0.428 (0.278)
u_sd_len_sum	0.009** (0.004)	0.009* (0.005)
cum_reviews	-0.001** (0.000)	-0.001** (0.000)
u_avg_rating	0.230*** (0.052)	0.225*** (0.053)
c_avg_len_sum	-0.025** (0.011)	-0.022* (0.012)
u_sd_len_sum	0.032** (0.013)	0.030** (0.013)
smart_con	0.708*** (0.149)	0.699*** (0.147)
easy	0.808*** (0.156)	0.806*** (0.154)
save	0.700*** (0.152)	0.713*** (0.154)
func	1.408*** (.264)	1.407*** (0.263)
support	1.148*** (0.224)	1.147*** (0.223)
price value	0.665*** (0.138)	0.675*** (0.139)
privacy	1.938*** (0.500)	1.915*** (0.496)
amazon	0.291* (0.162)	0.298* (0.162)
env	0.022 (0.686)	0.058 (0.698)
Z.u_avg_rating	-0.033* (0.019)	-0.032* (0.019)
Z.nest	0.544*** (0.174)	0.544*** (0.174)
Z.honey	0.401** (0.174)	0.398** (0.174)
Z.lux	0.382** (0.174)	0.386** (0.174)
Z.hunter	0.594*** (0.196)	0.582*** (0.196)
Z.venstar	0.175 (0.224)	0.229 (0.224)
Time Fixed Effect	Yes	Yes
LR tests, $X^2(6)$	30.60***	28.69***
Loglikelihood	-4,014.864	-4,003.836
AIC	8,143.728	8,159.673
BIC	8,518.593	8,659.494

* Notes: *P-value* = *$p < .1$; **$p < .05$; ***$p < .01$; only statistically significant variables and related variables are reported; standard deviation in parentheses.

footprints (DFs) and sentiment variables for each product content dimension.

Even though there is still the possibility of selection on unobservable factors, the models using DF and sentiment variables show a much better model fit than model_o1 and the same sign and similar coefficient magnitudes for key variables across the HETOP models. This similarity indicates the importance of digital footprint mining and sentiment analysis in estimating consumer preference.

V. EX ANTE PREDICTION USING MACHINE LEARNING

Increased ability to predict potential customers' level of satisfaction with a product would enable firms to better target potential positive consumers. Therefore, six different machine learning models (Appendix) are applied here to predict potential consumers' sentiment.

Classification is a prediction task for a discrete dependent variable (i.e., label). For example, predicting a five-star rating from online product reviews involves multiclass classification, which is often a more difficult task than binary classification. Bouazizi and Ohtsuki [34] showed that the accuracy of sentiment classification of a balanced dataset from Twitter decreased from 81.3% in a binary classification to 60.2% in a multiclass classification with seven different sentiment classifications.

Many scholars have simplified multiclass classification into a binary classification (positive or negative) [2], [35]. This study provides each classifier's performance in the

TABLE 7. Class distribution in the three-class classification.

Class	Total Set		Total Training Set		Sub Training Set		Valid Set		Test Set	
3	4,149	78.18%	3,911	78.16%	3670	78.04%	241	80.07%	238	78.55%
2	336	6.33%	322	6.43%	308	6.55%	14	4.65%	14	4.62%
1	822	15.49%	771	15.41%	725	15.42%	46	15.28%	51	16.83%
Total	5,307	100.00%	5,004	100.00%	4703	100.00%	301	100.00%	303	100.00%

three-class classification that contains "positive (3; five- and four- ratings)," "neutral (2; three- rating)," and "negative (1; two- and one- ratings)".

As shown in Table 7, the rating distribution in this study is skewed to the positive class, so it is an imbalanced dataset. Classification of imbalanced data is challenge in machine learning because classification results tend to be biased toward the majority class.

Class weighting is a popular approach to mitigate the imbalanced class problem [36]. In detail, class weighting puts more weight on the minority class (three-star ratings) than majority classes in a machine learning model's loss function, making the loss function more sensitive to the minority class and less sensitive to majority classes. In this study, class weighting is applied to each machine in this section as a hyperparameter.

The data used in these machine learning models is sampled from October 12, 2005 to July 17, 2014, and the total sample size is 5,307 reviews (and reviewers). This study defines the validation and test datasets with similar sample sizes (301 and 303 reviews, respectively) and time intervals (about a month). This study further assumes that the weather and seasonality are similar in the validation and test datasets.

The ex ante classification of potential reviewers' sentiment is divided into ex ante and partial ex ante classification. First, the ex ante classification is the prediction of potential consumers' sentiment before they make a purchase. In this case, firms do not know reviewers' ratings, reviews, or reviewed or purchased thermostats, so these ex post variables are excluded.

Second, the partial ex ante classification is a prediction of potential consumers' sentiment before they write a review for a purchased thermostat. In this case, firms know the types of thermostats that consumers have purchased. However, they do not know the consumers' rating and reviews for the purchased thermostats because the consumers have not posted a review yet. Therefore, reviewers' ratings and reviews are excluded from the partial ex ante model, but the programmable thermostat dummy variables are included in the partial ex ante model.

If the machine learning model is too closely fitted to the training data, the fitted model's prediction performance for new data points in the validation set will decrease. This modeling error is usually called overfitting in machine learning [37]. The optimal hyperparameter values for each prediction machine are selected when the optimal values mitigate the overfitting problems during the hyperparameter tuning process.

To avoid overfitting, the original dataset is split in the training step into a total training set and a test set, and the total training set is also divided into a training set and a validation set for hyperparameter tuning. Each machine learning model is trained on the training set and predicts new data points in the validation set. The optimal hyperparameter values are selected when the validation loss stops decreasing while the training loss keeps decreasing.

In the test set prediction step, each prediction model is also trained on the total training data with the optimal hyperparameters selected during the training step. The model trained on the total training data predicts the label in the test set. Reviewers' sentiment classification in the test set can be interpreted as predicting the strength of potential consumers' preferences.

A. MACHINE LEARNING MODELS FOR EX ANTE PREDICTION

The support vector machine (SVM) [38] and decision tree (DT) [39] models are base models (single classifiers) used to compare their prediction performance with more complex models.

Ensemble methods use a set of base classifiers. Dietterich [40] suggested that ensemble models often perform better than single classifiers because: (1) averaging classifiers may reduce the probability of using the wrong classifier; (2) different starting points for each classifier's optimization may reduce the possible local optima; and (3) combining classifiers may represent the correct function for mapping features to labels. Random forest (RF) [41] and extreme gradient boosting (XGB) [42] are tree ensemble models.

Recently, deep learning (DL) has shown dramatic progress in diverse areas. DL automatically learns a representation of data for required tasks [43]. The artificial neural net (ANN) [44] and long–short-term memory (LSTM) [45] models are DL models.

B. EX ANTE PREDICTION PERFORMANCE IN SENTIMENT CLASSIFICATION

The prediction performance criteria for sentiment classification are:

1. Accuracy: the ratio of the total number of correctly classified reviews over the total number of reviews;

2. Precision: the fraction of reviews correctly classified for a given star rating over the total number of reviews classified as the star rating;
3. Recall: the fraction of reviews correctly classified for a given star rating over the true number of reviews belong to the star rating; and
4. F-score: the weighted average of precision and recall in the following format:

$$F1\ score = 2 \times (precision \times recall)/(precision + recall)$$

According to the studies conducted by Ibrahim *et al.* [46] and Jeni *et al.* [47], the F1 score may be a better evaluation criterion for this imbalanced dataset because accuracy could mislead the prediction performance of classifiers for an imbalanced dataset. For example, if a machine learning model classifies all the instances in the test set (Table 7) as a positive class, the accuracy will be.7855 (the minimum reasonable accuracy of a classifier). Accordingly, the weighted average macro F1 score (WA F1) is the evaluation criterion for each model's prediction performance in this study as follows:

$$\text{WA F1} = \sum\nolimits_{k=1}^{K} \frac{N_k}{N} \times k\ class's\ F1 - score$$

C. EX ANTE PREDICTION PERFORMANCE IN THE THREE-CLASS SENTIMENT CLASSIFICATION

The predictive performance of six popular prediction machines with six different feature sets can be seen in Table 8. Model 1 (''at time model'') is the base model that contains only 37 observable variables. This model is a base model (feature set) for the prediction performance of the six machine learning algorithms with different models (i.e., different feature sets). Without digital footprints and sentiment variables, as in the case of model 1, only the prediction performance of SVM in the WA F1 score is slightly better than that of the econometric model (HETOP). In this case, there is no strong incentive to apply other complex machine learning models to predict potential consumers' sentiment instead of the base machine learning model (SVM) or conventional econometric model (HETOP). In addition, the predictive performance of machine learning and econometric models with this feature set is very low.

Models 2, 3, and 4 (Table 8) are ex ante models used to predict consumers' potential sentiment for PTs before they make a purchase. Model 3 (the ''ex ante sub-model'') shows the highest predictive performance of the best classifier among all six models (including the three ex ante models). RF and XGB in model 3 are not only the best prediction machine among the six classifiers in all six models with a WA F1 score of 0.74, but also shows the highest accuracy among the six classifiers in all six models with a score of 0.802 (Table 8).

Surprisingly, adding more price variables to model 3 does not improve the best classifiers (RF and XGB)' prediction performance in model 4. This result indicates that adding a potentially biased variable (price at the time of web scraping) to prediction machines may not improve the prediction performance.

Models 5 and 6 (Table 8) are ''partial ex ante'' models used to predict consumers' potential sentiment for the PTs purchased before they write a review. These models contain the product dummies for 71 PTs; therefore, firms know the type of PTs purchased by the consumers.

Surprisingly, adding these product dummies to the feature set in model 3 does not improve the WA F1 score of most of the classifiers (Table 8). Therefore, information about purchased PTs may not be very useful for improving classifiers' prediction performances in model 3.

Table 9 provides the detailed model structure in model 3, the optimal hyperparameters for each model, and the confusion matrix for each classifier's prediction. Notably, all the classifiers in model 3 show a zero WA F1 score for the minority class (2; three-star rating). This result shows the biased prediction problem in the imbalanced data. If a three-star-rating reviewer group is the minority group in a society, it may cause unfairness and inequality issues.

VI. SENTIMENT CLASSIFICATION USING NLP

Labeling text data for sentiment analysis often requires high-cost, time-consuming, and labor-intensive work. If the volume of review data is larger, the required time, labor, and financial cost for annotation will increase as well. In this case, firms can reduce these labeling costs by leveraging natural language processing (NLP).

Firms can apply deep learning methods to identify semantic meanings from review text. After training NLP models on an expert-annotated training dataset, the trained NLP models could classify the reviewers' sentiment toward a specific product content dimension (PCD) in a new review text dataset. Firms can apply these sentiment analyses to heuristic, fast, data-driven business decision making for better consumer support and feedback.

As a digital experiment for examining NLP's potential for sentiment analysis, diverse NLP methods are applied to classify reviewers' sentiment toward a specific product content dimension (functionality) because the functionality dimension contains the least imbalanced data among the nine PCDs for programmable thermostats (PTs). As shown in Table 10, the reviewers' sentiment regarding the functionality is distributed as follows: positive (3) with 41.70%, neutral (2) with 32.77%, and negative (1) with 25.53%. This dataset is relatively balanced compared with the previous datasets.

Word embedding is a way to map words to real vector space. Word embedding assumes that numerical vectors generated from review text contain the semantic information in the review text. High quality word embedding vectors are essential for sentiment classification performance. Three different word-embedding approaches are applied in this study to convert review text into numerical input vectors: (1) word frequency-based embedding, (2) word distribution-based embedding, and (3) context-based embedding.

In particular, transfer learning has shown success in different NLP tasks and has become an important approach in NLP [48]–[50]. Transfer learning assumes that, when the

TABLE 8. Ex ante prediction results in the three-class classification (Appendix).

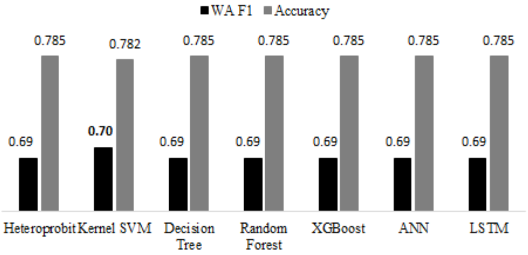
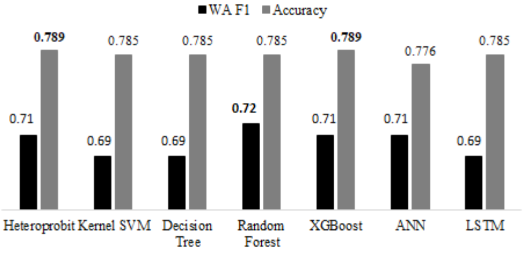
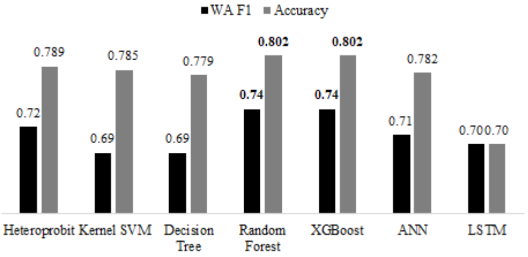
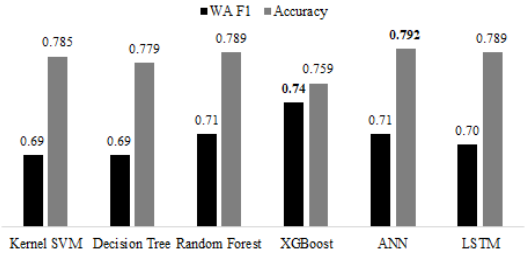
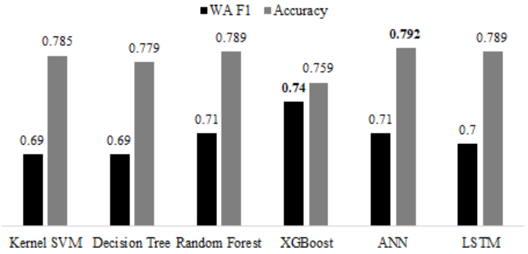
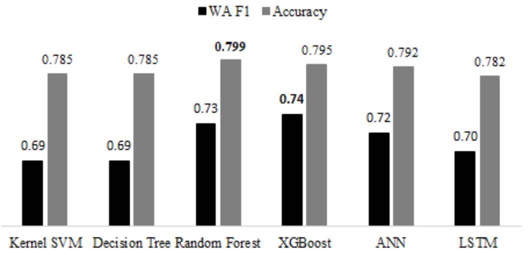
Models	Variables	Weighted Average Macro F1-score and accuracy
Model 1: at time model	37 variables including: 1. variables when the reviewers write a review 2. time fixed effects	
Model 2: ex ante model	59 variables including: 1. 37 variables from at time model 2. 22 DFs variables including the reviewer's volume of prior reviews in all categories	
Model 3: ex ante-sub-model	90 variables including: 1. variables from 'At time' model 2. 32 variables for the reviewer's volume of prior reviews in each sub-category	
Model 4: ex ante-sub-price model	94 variables including: 1. 90 variables from ex ante-sub-model 2. price and price DFs (4 variables)	
Model 5: partial ex ante-sub-model	161 variables including: 1. 90 variables from ex ante-sub-model 2. 71 product dummies	
Model 6: partial ex ante-sub-price model	165 variables including: 1. partial ex ante-sub- model (161 variables) 2. price and price DFs (4 variables)	

TABLE 9. Three-class classification: Ex ante-sub model (90 Variables).

Models	Hyperparameter	Accuracy	Precision	Recall	F1-score	Confusion matrix			
							1	2	3
Heteroprobit		0.789	1: 0.56 2: 0.00 3: 0.80 WA: 0.72	1: 0.10 2: 0.00 3: 0.98 WA: 0.79	1: 0.17 2: 0.00 3: 0.88 WA: 0.72	1	5	0	46
						2	0	0	14
						3	4	0	234
							1	2	3
Kernel SVM	Kernel: RGB Gamma: 1.0 C: 0.1	0.785	1: 0.00 2: 0.00 3: 0.79 WA: 0.62	1: 0.00 2: 0.00 3: 10.00 WA: 0.79	1: 0.00 2: 0.00 3: 0.88 WA: 0.69	1	0	0	51
						2	0	0	14
						3	0	0	238
							1	2	3
Decision Tree	Criteria: entropy Max depth: 4	0.779	1: 0.25 2: 0.00 3: 0.79 WA: 0.66	1: 0.02 2: 0.00 3: 0.99 WA: 0.78	1: 0.04 2: 0.00 3: 0.88 WA: 0.69	1	1	0	50
						2	0	0	14
						3	3	0	235
							1	2	3
Random Forest	Tree numbers: 16 Depth: 42	0.802	1: 0.73 2: 0.00 3: 0.80 WA: 0.75	1: 0.16 2: 0.00 3: 0.99 WA: 0.80	1: 0.26 2: 0.00 3: 0.89 **WA: 0.74**	1	6	2	43
						2	0	0	14
						3	5	0	233
							1	2	3
Xgboost	Tree number: 100 Depth: 4 Learning rate: 0.2	0.802	1: 0.78 2: 0.00 3: 0.80 WA: 0.76	1: 0.14 2: 0.00 3: 0.99 WA: 0.80	1: 0.23 2: 0.00 3: 0.89 **WA: 0.74**	1	7	0	44
						2	0	0	14
						3	2	0	236
							1	2	3
ANN	Epoch: 3 Drop out: .4 Learning rate: 0.0002 Hidden layer 1 node:180 Hidden layer 2 node:180	0.782	1: 0.38 2: 0.00 3: 0.79 WA: 0.69	1: 0.06 2: 0.00 3: 0.98 WA: 0.78	1: 0.10 2: 0.00 3: 0.88 WA: 0.71	1	3	0	48
						2	1	0	13
						3	4	0	234
							1	2	3
LSTM	Epoch: 232 Learning rate: 0.0002 Hidden layer node: 322	0.700	1: 0.50 2: 0.00 3: 0.79 WA: 0.70	1: 0.02 2: 0.00 3: 10.00 WA: 0.79	1: 0.04 2: 0.00 3: 0.88 WA: 0.70	1	1	0	50
						2	0	0	14
						3	1	0	237

* Note: WA indicates weighted average macro values; the horizontal labels from 1 (left) to 3 (right) are the predictive classes, while the vertical labels from 1 (top) to 3 (bottom) are the true classes. The values on the diagonal are the number of correct predictions for the classes mapped to the horizontal or vertical classes.

training dataset is relatively small, using parameters in pre-trained models trained with big data could improve NLP models' performance in a new task.

Two popular transfer learning approaches are fine-tuning [48] and further pre-training [51]. The fine-tuning approach simply reuses a pre-trained model for new target tasks. A further pre-training approach involves training a pre-trained model with domain data to update the weights in the pre-trained model to reflect contextual domain information. The fine-tuning and further pre-training methods are applied to the W2V and BERT models in this study.

TABLE 10. Sentiment distribution in the functionality dimension.

Class	Nest	Honeywell	Lux	Hunter Fan Com	Venstar	White Roger	Total	Percent
3	789	744	555	70	43	12	2,213	41.70%
2	700	619	315	46	32	27	1,739	32.77%
1	584	424	269	45	18	15	1,355	25.53%
Total	2,073	1,787	1,139	161	93	54	5,307	100%

On top of each word-embedding vector generated from the review text, tree-based ensemble models (RF, XGB) and a deep learning model (CNN) are applied to classify reviewers' sentiment toward the functionality dimension. Each classification model is combined with a suitable word-embedding method for each classifier's characteristics.

A. WORD EMBEDDING: MAPPING TEXT TO NUMERICAL VECTORS

Frequency-based embedding is a simple way to map each review text to numerical vectors. Term frequency–inverse document frequency (TF-IDF) is a frequency-based type of word embedding and penalizes the high-frequency words in the entire review [35]. On top of the TF-IDF embedding vectors from the review text data, RF and XGB are applied for sentiment analysis. TF-IDF has a high-dimensional spare matrix and cannot represent similarity, ambiguity, and contextual meaning in a text (Appendix).

The Word2Vec (W2V) [52] model is a word distribution-based embedding method and generates dense embedding vectors representing each word's semantic meaning. For example, the W2V model may generate similar embedding vectors for ''pen'' and ''pencil'' because the two words contain similar semantic meanings. In this study, the W2V model is trained with all the reviews (N = 1,926,047) in the ''tool and home improvement'' category and the number of unique words is 73,856. The hyperparameters are the W2V embedding dimension, window size, and training dataset. After hyperparameter tuning, the optimal W2V embedding dimension is 100 and the optimal window size is 5 (Appendix).

Bidirectional Encoder Representations from Transformers (BERT) is a state-of-the-art context-based embedding method. BERT can represent the same word in a sentence with different embedding vectors by reflecting the contextual meaning of each word in the sentence. For example, in the sentences ''I did not like this thermostat in the past. Now, I love this thermostat,'' the word ''thermostat'' occurs twice, in the first and in the second sentence. BERT generates different embedding vectors for ''thermostat'' in the first and second sentences based on the contextual information in them. Meanwhile, context-free embedding models (e.g., TF-IDF and W2V) generate the same embedding vectors for ''thermostat'' in both sentences.

In particular, the domain expert in this study reads and annotates all 5,307 reviews for PTs and finds that the review text often contains a comparison between the previously owned PT and the newly purchased PT, so the same word in the review often represents different contexts based on its position in the review. For example, ''I disliked the previous thermostat. However, I love this new thermostat.'' In this text, even though the word ''thermostat'' occurs both in the first and in the second sentence, the first one may contain a negative sentiment and the second one may contain a positive sentiment. However, context-free embedding models cannot capture different semantic meanings of the same word in different positions in the review sentences. In contrast to the context-free embedding models, BERT (context-based embedding) can find the contextual difference between occurrences of the same word in different positions in the review sentences.

This study uses the BERT-based model, which contains 30,522 unique tokens with 768 embedding dimensions for fine-tuning and further pre-training. With a fine-tuned BERT, the convolutional neural network (CNN) is applied on top of the pre-trained embedding from the original BERT model. Having further pre-trained BERT, the BERT embedding is updated by training on the review text data and is used as input vectors for the CNN classifier. Recently, Gururangan *et al.* [51] and Sun *et al.* [53] showed that further pre-training with domain data could improve machine learning models' performance.

B. CONVOLUTIONAL NEURAL NETWORK (CNN) FOR SENTIMENT CLASSIFICATION

Many studies have applied a CNN for text classification and shown good performance [54]–[56]. Liu *et al.* [1] and Timoshenko and Hauser [2] applied CNN text classification on top of W2V embedding trained on review data. In this study, the CNN classifier on top of BERT or W2V embedding is applied for sentiment analysis (Appendix).

According to Zhang and Wallace [56], the filter size and the number of filters are key hyperparameters for a CNN model where a 1-max pooling is better than other pooling methods, and regularization has little influence on the performance of the CNN classification. This study applies multiple feature sizes and different filters to find the optimal parameters. Input embedding vectors are generated from multiple versions of the W2V and BERT models. For structured data, 161 variables are selected from the partial ex ante sub-model as input variables for the full model (text and structured data model).

TABLE 11. Class distribution in the functionality dimension.

	Total Set		Total Training Set		Training Set		Valid Set		Test Set	
Class	Count	Shares	Count	Shares	Count	Share	Count	Share	Count	Share
3	2,213	41.70%	2,098	41.93%	1969	41.87%	129	42.86%	115	37.95%
2	1,739	32.77%	1,625	32.47%	1523	32.38%	102	33.89%	114	37.62%
1	1,355	25.53%	1,281	25.60%	1211	25.75%	70	23.26%	74	24.42%
Total	5,307	100.00%	5,004	100.00%	4,703	100.00%	301	100.00%	303	100.00%
Period	Oct 12, 2005 – July 17 2014		Oct 12, 2005 – May 17, 2014		Oct 12, 2005 –Mar 16, 2014		Mar 17, 2014 –May 17 2014		May 18, 2014 – July 17 2014	

C. SENTIMENT CLASSIFICATION EXPERIMENT DESIGN

Table 11 shows the distribution of the three classes in the functionality decision in the review text. This dataset is relatively less imbalanced than the previous datasets, so the prediction performance of the minority class (1) may be better than in previous cases. In Table 11, the bottom line of the test set accuracy is 0.3795.

This study defines the partial and full models based on the type of features in the model. The partial model simplifies the feature engineering by excluding digital footprint (DF) mining from user-generated content (UGC) to generate numerical input variables. In general, DF mining requires intensive manual coding and adequate computing resources (e.g., mass storage space and big-memory computers). Generating input variables from DFs also requires a large online product review dataset that contains individual user IDs, product IDs, and time stamps. Firms often want to reduce feature engineering by focusing only on review text data (the partial-model approach). However, the full-model approach shows how to combine unstructured review text data with structured data to improve a classifier's performance.

In this section, tree ensemble models (RF and XGB) are selected as baseline models to compare their prediction performance with more complex models. The TF-IDF embedding method is applied to the RF and XGB models because these models are incompatible with the two-dimensional word-embedding vectors generated by the W2V and BERT models.

The CNN model is a popular deep learning model for text classification. In particular, various CNN models on top of BERT or W2V embedding vectors are the main classifiers in this section. In this study, the CNN model's hyperparameters are the length of the review text, training epochs, number of filters, filter sizes, dropout rate, and learning rate.

The W2V embedding models are trained on different review datasets with different window sizes and embedding dimensions. The CNN classifier on top of Google's pre-trained W2V embedding (trained on three million words and phrases from Google News) shows lower prediction performance than the CNN classifier on top of W2V embedding generated in this study (trained on online product review data from Amazon). In particular, two different online product review datasets are used for training the W2V models: (1) W2V_S (N = 169,809 reviews), containing all reviews of the target reviewers across all categories over the entire sample period; and (2) W2V_L (N = 1,926,047 reviews), consisting of all reviews in the ''tool and home improvement category'' over the entire sample period. The W2V model trained on W2V_L shows better performance for sentiment analysis in this section than the W2V model trained on W2V_S and on Google's pre-trained model.

The BERT models are applied to word-embedding methods with two different approaches, the fine-tuning and further pre-training approaches. The fine-tuning approach simply reuses the pre-trained embedding vectors from the original model as input-embedding vectors for a classifier. This approach relies on transferring learning and has recently been shown to be successful in the performance in NLP tasks.

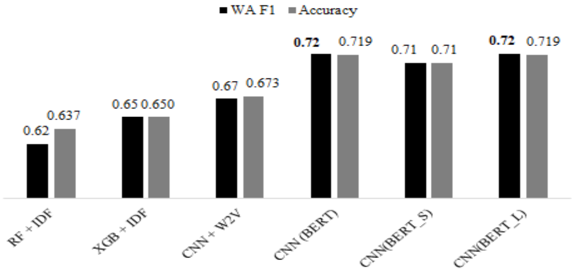
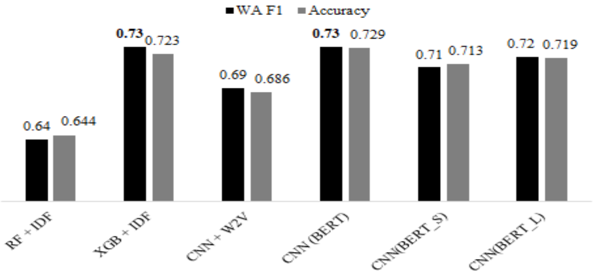
A further pre-training approach updates the pre-trained embedding vectors by training the pre-trained model on domain data to adapt domain context information to embedding vectors. However, there is no ground truth or theoretical proof supporting the assumption that further pre-training ensures better performance with noisy online product review data. Two different online product review datasets receive further pre-training: (1) BERT_S (N = 169,809 reviews), containing all reviews of the target reviewers across all categories over the entire sample period; and (2) BERT_L (N = 1,926,047 reviews), consisting of all reviews in the ''tool and home improvement category''.

For further pre-training of the BERT model on domain-specific review data, the hyperparameters are the learning rate, batch size, and further training steps. In this study, the optimal hyperparameters for further training BERT are learning rate 0.00001, batch size 32, and 1,926,047 training steps. In the BERT model, the maximum length of tokens is fixed as 512 (510 without special tokens); therefore, 512 is the maximum length of review tokens for the BERT model in this study.

D. SENTIMENT CLASSIFICATION RESULTS

Table 12 presents the results of the sentiment classification of reviews about a specific product content dimension. The classification models are divided into the partial model (using text only) and the full model (using text and structured data).

TABLE 12. Sentiment classification results.

Models	Word embedding	Weighted Average Macro F1-score and Accuracy
Partial model (Text only)	TF-IDF: embedding on 5,307 reviews W2V: embedding on W2V_L (dimension: 100, size: 5) BERT: pre-trained embedding BERT_S: further pre-trained embedding on BERT_S BERT_L: further pre-trained embedding on BERT_L	
Full model (Text + partial ex ante-sub-model)	Word embedding conditions in 'Text only' and 161 variables from the partial ex ante-sub-model	

In the partial model, the CNN models on top of fine-tuned BERT or further pre-trained BERT_L embedding show the highest WA F1 score and accuracy. Accuracy is an important evaluation metric for measuring the prediction performance because the dataset in this section is relatively more balanced than the datasets in the previous sections.

All the CNN models on top of BERT embedding shows better prediction performance than the tree ensemble models and the CNN models on top of context-free embedding (TF-IDF and W2V embedding). This result indicates that BERT is a better embedding method for sentiment classification.

It demonstrates that the identification of contextual information from review text is a critical factor for the sentiment classification of online product reviews (Table 12).

In the full model, the CNN model on top of the fine-tuned BERT embedding shows the highest WA F1 score and accuracy (Table 12). This result indicates that firms can easily implement sentiment analysis without intensive training steps for word-embedding models and accomplish high prediction performance by reusing pre-trained BERT embedding as input embedding vectors. The CNN models with further trained BERT embedding show lower prediction performance than the CNN model with pre-trained BERT embedding. Therefore, further pre-training of BERT may not be a suitable embedding method in this case.

Surprisingly, the class-weighted XGB on top of TF-IDF embedding shows the same WA F1 score as the CNN on top of pre-trained BERT embedding (Table 13). The prediction performance of XGB with text and structured data is higher than that of XGB with text data only. This result may be due to the weighted XGB's good prediction performance with structured numerical variables.

In contrast to the previous sections, the dataset in this section is relatively balanced, so the imbalanced classification problem is not a critical issue in this section and the classification performance for the minority class is not low. Overall, the CNN on top of fine-tuned BERT is the best option in all cases, with high prediction performances and low computational costs for training the embedding model. In addition, the full-model cases are mostly superior to the partial-model cases.

VII. CONCLUSION

This study finds that all HETOP models containing DFs and sentiment variables show a higher model fit than the base model containing no DFs or sentiment variables. Furthermore, machine learning models containing DFs and sentiment variables show better prediction performance than the base model. These points indicate the importance of DF mining and sentiment analysis for estimation and prediction tasks.

The HETOP models' results show that a consumer is less likely to give a five-star rating for a reviewed programmable thermostat (PT) if he or she: (1) writes a longer review summary and body, (2) has a lower variance of review summary length in prior reviews, a larger volume of prior reviews across all categories, and a higher average rating in prior reviews across all categories, (3) writes a review for the PT that has a higher average length of review summary and/or lower variance of review summary length in prior reviews, (4) writes a larger volume of prior reviews in specific product categories.

The eight sentiment variables positively affect the probability of a 5-star rating. The sentiment variables represent the target consumers' sentiment toward product content dimensions (PCDs). The dimensions are (1) smart connectivity,

TABLE 13. Sentiment classification results.

Models	Hyperparameter	Accuracy	Precision	Recall	F1	Confusion matrix			
RF on top of TD-IDF	Tree numbers: 29 Depth: 26	0.644	1: 0.76 2: 0.65 3: 0.59 WA:0.66	1: 0.53 2: 0.67 3: 0.70 WA: 0.64	1: 0.62 2: 0.66 3: 0.64 WA:0.64		1	2	3
						1	39	12	23
						2	6	76	32
						3	6	29	80
XGB on top of TD-IDF	Tree number: 100 Depth: 7 Learning rate: 0.2 Class weighted*	0.723	1: 0.84 2: 0.74 3: 0.67 WA:0.74	1: 0.80 2: 0.66 3: 0.77 WA: 0.73	1: 0.82 2: 0.69 3: 0.72 **WA: 0.73**		1	2	3
						1	59	5	10
						2	6	75	33
						3	5	22	88
CNN on top of W2V*	Max length: 1800 Epoch: 22 Number of filters:200 Filter size: (3,4,5) Dropout: 0.7 Learning rate: 0.0001	0.686	1: 0.81 2: 0.65 3: 0.65 WA:0.70	1: 0.74 2: 0.62 3: 0.71 WA:0.69	1: 0.77 2: 0.64 3: 0.68 WA: 0.69		1	2	3
						1	55	12	7
						2	6	71	37
						3	7	26	82
CNN on top of BERT	Max length: 512 Epoch: 15 Number of filters: 200 Filter sizes: (2,3,4) Dropout: 0.7 Learning rate: 0.00001 Class weighted*	**0.729**	1: 0.94 2: 0.63 3: 0.78 WA:0.76	1: 0.62 2: 0.58 3: 0.68 WA:0.73	1: 0.75 2: 0.72 3: 0.73 **WA:0.73**		1	2	3
						1	46	21	7
						2	2	97	15
						3	1	36	78
CNN on top of BERT further pre-training (BERT_S*)	Max length: 512 Epoch: 49 Number of filters:200 Filter sizes: (2,3,4) Dropout: 0.6 Learning rate: 0.00001	0.713	1: 0.88 2: 0.64 3: 0.74 WA:0.73	1: 0.68 2: 0.84 3: 0.61 WA:0.71	1: 0.76 2: 0.72 3: 0.67 WA: 0.71		1	2	3
						1	50	14	10
						2	3	96	15
						3	4	41	70
CNN on top of BERT further pre-training (BERT_L*)	Max length: 512 Epoch: 11 Number of filters:300 Filter sizes: (3,4,5) Dropout: 0.7 Learning rate: 0.0001	0.719	1: 0.73 2: 0.69 3: 0.75 WA:0.72	1: 0.77 2: 0.72 3: 0.69 WA:0.72	1: 0.75 2: 0.70 3: 0.71 WA:0.72		1	2	3
						1	57	11	6
						2	11	82	21
						3	10	26	79

* Notes: class weight: class [1, 2, 3], weights for each class [1.2945, 1.0293, 0.7962]; W2V: trained on W2V_L and embedding dimension is 100 with window size 5; BERT further training on BERT_S: further pre-trained with target reviewers' reviews across all categories (169,809 reviews) and further pre-trained with 849,045 steps (5 epochs with 169,809 steps per epoch); BERT further training on BERT_L: further pre-trained with all reviews in the "tool and home improvement" category and further pre-trained with 1,926,047 reviews.

(2) easiness, (3) energy and money saving, (4) functionality, (5) support, (6) perceived price value, (7) privacy, and (8) the Amazon effect. The results suggest that consumers consider not only the smartness of programmable thermostats but also the easiness of using the device. Surprisingly, consumers also consider the value of privacy. Without extracting the latent product content dimension from the online product reviews, firms may not be able to discover these latent factors that affect consumer preferences. To the best of the author's knowledge, this is the first study to address the effect of the online retail market platform's service quality on the consumers' star ratings. Without consideration of the online platform service quality effect, empirical results will be biased. This approach can be applied to design the promotion of products, measure the effects of policies (such as energy star certification) on consumers' preferences in the online retail market platform, and identify the factors that affect consumer satisfaction or dissatisfaction.

This study also finds that extreme gradient boosting (XGB) is the best prediction machine among six popular machine learning algorithms for predicting individual consumers' sentiment before they make a purchase or write a review. In addition, this study shows how to combine variables generated from text and other numerical variables to make predictions. This study also shows each machine learning algorithm's performance in sentiment classification with the imbalanced dataset, finding that all the machine learning algorithms show low prediction performance for the minority class. The imbalanced classification problem can cause social inequality or unfairness issues if the majority class group belongs to the minority groups in a society. Above all, this approach can be implemented in an online review platform to design better target marketing strategies and recommendation systems.

This study applies natural language processing (NLP) to classify the target consumers' sentiments toward a specific product content dimension from the review text. Firms can apply this approach to reduce expensive domain expert annotation costs and implement data-driven business decisions. This approach provides empirical evidence that the context-based embedding (BERT) approach outperforms context-free embedding models (TF-IDF and Word2Vec). In particular, this study applies transfer learning concepts by applying pre-trained BERT embedding as input embedding for the CNN classifier. It also suggests that the further pre-training of BERT with domain review text data may not guarantee the improvement of prediction performance.

In sum, the approaches in this study are interpretable, applicable, and scalable to a wide range of goods, allowing for the identification and prediction of unobserved consumer preferences and sentiments associated with product content dimensions for a specific target product group.

Applying the approaches in this study to specific search goods (e.g., organic or non-organic milk) or credible goods (e.g., wine) will be a good extension of this study. The effects of expensive domain expert annotation and relatively inexpensive crowdsourcing annotation (e.g., Amazon Mechanical Turk) for sentiment classification performance will also be a valuable topic for future research. In addition, a study that examines true and fake reviews on different online platforms will be useful for identifying the differences between true and fake reviews.

APPENDIX A
CONCEPTUAL FRAMEWORK

The conceptual consumer space shows the segmentation of consumers (Figure 2). The purpose of this consumer space concept is to derive the group of consumers who become reviewers on Amazon.

The total consumer group is denoted as S_t. This total group is divided into two groups, those who are users of Amazon, S_a, and those who are not, S_{na}. This study assumes that members of the non-Amazon user group S_{na} do not write and read reviews on Amazon.

The Amazon user group S_a is split into two subgroups, those who write reviews, S_{aw}, and those who do not, S_{anw}. It should be noted that even though consumers in S_{aw} write reviews, it is possible that their review data contains bias. Accordingly, this study assumes these biased reviews reduce the credibility of the information found in the reviews.

Above all, if a researcher analyzes the review data written by the consumer group S_{aw} is analyzed and used to estimate and predict individual consumer preferences of the entire Amazon user group S_a, it will cause sample selection bias because there is no information about S_{anw}. Therefore, this study aims to estimate and consumer preferences for the group of Amazon users who write a review, i.e., S_{aw}, by using the review data written by this group (S_{aw}) while excluding biased reviews (from the subgroup S_{awb}). Consequently, this paper implements specific pre-processes to remove the reviews written by S_{awb}.

In addition, this study extracts individual reviewers' DFs for a specific product group from a dataset of 141 million Amazon reviews. The DFs are divided into two groups.

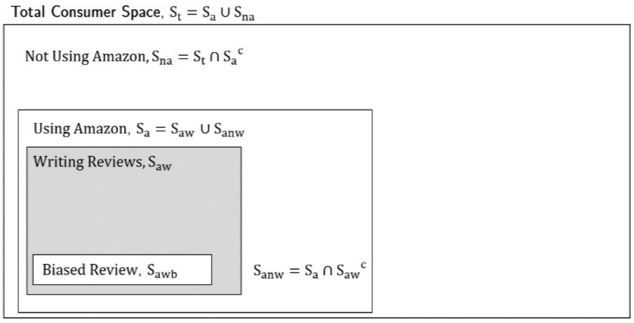

Total Consumer Space, $S_t = S_a \cup S_{na}$
Not Using Amazon, $S_{na} = S_t \cap S_a^c$
Using Amazon, $S_a = S_{aw} \cup S_{anw}$
Writing Reviews, S_{aw}
Biased Review, S_{awb}
$S_{anw} = S_a \cap S_{aw}^c$

FIGURE 2. Total consumer space.

1.User DFs: reviewer i's DFs before writing a review of thermostat p on day t_i.

$$\sum_{t_i^a}^{t_i^b} df_{ipt_i}(\cdot),\ \text{ where } t_i^a = \underset{t_i^a}{\operatorname{argmax}} |t_i - t_i^a| \text{ and } t_i > t_i^a$$

$$t_i^b = \underset{t_i^b}{\operatorname{argmin}} |t_i - t_i^b| \text{ and } t_i > t_i^b \geq t_i^a$$

$df_{ipt_i}(\cdot)$ is a DF function for reviewer i who writes a review of p before t_i.

2. Crowd DFs: the crowd's (other prior reviewers') DFs for thermostat p before i writes a review of thermostat p on day t_i.

$$\sum_{j\neq i}^{J}\sum_{t_j^a}^{t_j^b} df_{jpt_j}(\cdot),\ \text{ where } \{\forall J \in R \text{ and } 1 \leq j \leq J < \infty | i, t_i, p\}$$

$$t_j^a = \underset{t_j^a}{\operatorname{argmax}} |t_i - t_j^a| \text{ and } t_i > t_j^a$$

$$t_j^b = \underset{t_j^b}{\operatorname{argmin}} |t_i - t_j^b| \text{ and } t_i > t_j^b \geq t_j^a$$

APPENDIX B
DATA PRE-PROCESSING (DETAILED)

The Amazon review data used in this study are secondary [8]. The dataset has 142.8 million reviews that generated from May 1996 to July 2014. This data set does not have duplicate reviews for the same products. Detailed descriptions for each data pre-processing step are shown below:

Step 1: Selecting reviews with no missing values

The programmable thermostats (PTs) belong to the "tools and home improvement" category. Clarifying a specific product group (programmable thermostats) based only on the category may lead to noisy or missing samples. Therefore, the set of programmable thermostats is carefully defined through the following processes:

1. Selecting the category to which the product belongs from the following list.
   [["Tools & Home Improvement", "Building Supplies", "Heating & Cooling", "Thermostats & Accessories", "Thermostats", "Programmable']].

2. Removing the products that contain "non-programmable" in the title.
3. Selecting the products that contain "programmable" in the product description.
4. Removing the products that contain "non-programmable", "non programmable", or "programmable no" in the product description.
5. Removing the products that have a missing value in the brand or price variables.
6. Evaluating the image of each product to verify the robustness of the product set.

The PT set without missing values in either brand or price variables will henceforth be called "programmable thermostats." There are 110 thermostats in this set. Although the total number of initial reviews of the 110 PTs was 8,817, the total number of reviewers was 8,694, because some reviewers wrote multiple reviews.

This study considers only inexperienced consumers' first review of the PTs, because inexperienced consumers may become experienced consumers after they have written their first review. Second and third reviews of PTs from the same reviewer are deleted. Therefore, the total number of reviews of PTs used in this research is 8,694, the same as the number of reviewers.

Step. 2: Cleaning "suspicious one-time reviewers" and "always-the-same-rating reviewers"

Step 2.1 Cleaning "suspicious one-time reviewers"

Zhao *et al.* [18] indicated that fake reviews increase consumers' uncertainty about products and that more believable online reviews of experience goods have a larger effect on consumer choice. Some firms may write positive reviews about their products and negative ones about their rivals' products [14], [17], [19]. Accordingly, deleting potential fake reviews is essential to improve the credibility of review and reduce consumer uncertainty.

Mayzlin, Dover, and Chevalier (2014) defined the "suspicious reviewer" as one who writes a review for a hotel for the first time only during the sample period (October 2011) and showed that t3heir rating distribution is more polarized than that of the entire sample [14]. This study takes this into account by accessing individual reviewers' prior reviews in different categories over the entire sample period, defining a "suspicious one-time reviewer" as one who writes only a review for a PT as a first review and does not write reviews for any other products over the entire sample period.

This cleaning process assumes that suspicious one-time reviewers are less likely to write reviews of other products in different categories, excluding specific target product groups (own products or other competitors in the same product group), to minimize costs. In other words, suspicious one-time reviewers may be unlikely to post reviews outside of their product area. It is possible that they are actual reviewers. However, it is still reasonable to delete potential suspicious one-time reviewers to remove possible bias. In addition, suspicious one-time-reviewers do not have any digital footprints (DFs); therefore, these reviewers are supposed to be deleted in step 3 (deleting reviewers and reviews for products with no DFs.) A total of 1,165 reviews for 80 PTs are detected, written by 1,165 suspicious one-time reviewers.

Step 2.2 Cleaning "always-the-same-rating reviewers (ASRs)"

Some reviewers always give a star-rating at the same level for all reviewed products in all categories, regardless of the product quality. Such reviewers may not respond to product quality and previous reviews written by the crowd. Consequently, these reviews do not reflect the product quality. It may also be possible that the reviewers give the same rating level because the number of reviews is simply small. Over the sample period, 1,970 reviewers rated products in all categories at the same level; however, 1,165 reviewers wrote only 1 review and 316 reviewers wrote 2 reviews.

In this study, an "always-the-same-rating reviewers (ASR)" is a reviewer who writes more than 8 reviews with the same rating level. In detail, "Programmable thermostats" belong to "tool and home improvement" category in the Amazon review system. The majority rating in this category is a 5-star rating with a probability of 0.595. If the probability of the majority star rating in the five-scale star-rating system is 0.595 (extreme and subjective assumption), the probability that a reviewer independently writes reviews with the same majority star rating in nine consecutive reviews is 0.00934 (less than 1%). Only 69 reviewers write more than 8 reviews at the same star rating level (5 stars), surprisingly designating them as "always happy reviewers (AHRs)"; these 69 reviews for 25 PTs are removed.

There is no overlap between 1,165 suspicious 1-time reviewers and 69 ASR reviewers. The number of reviewers become 7,460 after removing 1,234 reviewers. As can be seen in Figure 3, the share of 1-star ratings of suspicious reviewers (18.9%) is about twice as large as that of reviewers after cleaning the suspicious 1-time reviewers and ASRs (9.69%). Therefore, there is potential for negative promotional reviews in the suspicious 1-time reviewers' reviews.

Step. 3: Deleting reviewers and reviews for products with no digital footprints (DFs)

Without DFs, it is impossible to measure the effect of DFs on a reviewer's rating for a PT when the reviewer writes a review for a PT for the first time. Accordingly, this procedure is followed: (1) 1,965 reviewers do not have any previous reviews of other products excluding PTs in all categories before the first day of writing a review for PTs; (2) 91 reviewers write a review for a PT that does not have any previous reviews written by other prior reviewers. The overlap between the 1,965 reviewers and the 91 reviewers is 28 reviewers; therefore, 1,234 reviewers are removed.

Step. 4: Selecting the top 6 major brands

This procedure restricts the reviewers who write a review for 6 brands that have more than 50 reviews. After this restriction, 5,307 reviewers write a review for the 6 major players, specifically Nest (2,073, 39.06%), Honeywell (1,787, 33.67%), Lux(1,139, 21.46%), the Hunter Fan Company (161, 3.3%), Venstar (93, 1.75%), and

White Roger (54, 1.02%). Finally, the number of reviewers and reviews for 71 PTs is 5,307.

Step 5: Identifying five latent product content dimensions in the reviews using LDA

Step 5.1: What is LDA (Latent Dirichlet allocation)?

LDA is a Bayesian unsupervised learning model used to identify latent topics in each review and the distribution of these topics in each review. The terminology for LDA in this study is defined as follows:

- $w_{i,n}$ is the nth word in the ith review and it follows a multinomial distribution.
- V (vocabulary) is the total number of unique words in the set of all review data
- K is the total number of topics in each review and is a hyperparameter
- The ith review is a sequence of N words as $r_i = (w_{i,1}, \ldots, w_{i,N})$
- A corpus is a set of M reviews as $R = (r_1, \ldots, r_M)$

As a generative probabilistic model, LDA assumes that each review is represented as a distribution over K topics as θ_i. θ_i is a vector in R^K that represents the proportion of each topic in the ith review. θ_i follows a Dirichlet distribution that has α as a Dirichlet parameter. In addition, φ_k is the kth topic vector in R^V that represents the proportion of each word that belongs to V in the kth topic. φ_k follows a Dirichlet distribution that has β as a topic hyperparameter. $z_{i,n}$ is a vector in R^K that maps the nth word in the ith review to topic k. $z_{i,n}$ and $w_{i,n}$ follow a multinomial distribution. Overall, θ_i, φ_k, and $z_{i,n}$ are latent variables and $w_{i,n}$ is an observable variable.

In addition, LDA assumes that w_R (words in reviews) is generated from the joint distribution of θ_R (the review's topic distribution) and φ_K (the topic's word distribution). The joint distribution indicates the word generation process in reviews as follows:

$$p(\varphi_K, \theta_R, z_R, w_R|\alpha, \beta) = \prod_{k=1}^{K} p(\varphi_K|\beta) \prod_{i=1}^{R} p(\theta_i|\alpha) \sum_{n=1}^{N} p(z_{i,n}|\theta_i) p(w_{i,n}|\varphi_k, z_{i,n}|\theta_i)$$

Excluding $w_{i,n}$, the other variables are latent variables. During the training process of LDA, the optimal values of the latent variables maximize the posterior probability. The posterior probability is denoted as follows:

$$p(\varphi_K, \theta_R, z_R|w_R) = \frac{p(\varphi_K, \theta_R, z_R, w_R)}{p(w_R)}$$

However, the denominator of the posterior probability is intractable for exact inference because φ_K, θ_R, and z_R are unobserved variables. In fact, various approximate inference methods are applicable for estimating posterior probability such as variational inference and Gibbs sampling.

Step 5.2: LDA Application in This Study

LDA is often called topic modeling. Topics in online product reviews indicate the product content dimensions for the products. The product review text for a specific product group contains finite product content dimensions (topics of product reviews) for the product group. Based on the empirical results of the LDA model and the theory [57], Liu *et al.* [2] divided the product content dimension for products from the online product review text into six dimensions as (1) esthetics, (2) conformance, (3) durability, (4) feature, (5) brand, and (6) price.

Though the theoretical framework is useful in general, this paper uses the LDA model to define the product content dimensions in online product reviews for a specific target product group (programmable thermostats) instead of the general category of goods.

After pre-processing, the number of unique words in 5,307 reviews (the review summary and the body of the review) for LDA is 4,554. The LDA model in this study contains 5 topic dimensions (Table 14). The number of optimal topics is determined by the coherence score (Figure 5) [58]. As can be seen in Table 14, the author, who is a domain expert in the power industry interprets, 5 subjective product content dimensions.

TABLE 14. Topics in reviews after LDA.

Topic	Interpretation	Top 15 keywords
1.Connectivity	The review describes WiFi, including wireless connection issues with software (e.g., App) and hardware (e.g., Heating, ventilation, and air conditioning).	wire, WiFi, power, device, connected, connect, wireless, Issue, common, update, app, router, software, hvac, connection,
2. Easiness	The review mentions ease of use, including simplicity of installation, programming, and use.	easy, work, install, program, installation, instruction, installed, simple, programming, nice, programmable, well, took, product, set
3. Saving	The review talks about energy savings, including money savings by reducing energy consumption.	energy, control, save, away, money, saving, heater, month, app, bill, iphone, electric, temperature, feature, best
4. Setting	The review contains content related to setting and control, and information related to temperature, time, scheduling, heating, and other devices.	temperature, time, set, heat, turn, day, back, go, temp, setting, system, need, want, work, change
5. Support	The review focuses on consumer support services before, during, and after they make a purchase.	support, customer, call, product, service, called, tech, told, said, company, hvac, issue, worked, working, customer_service

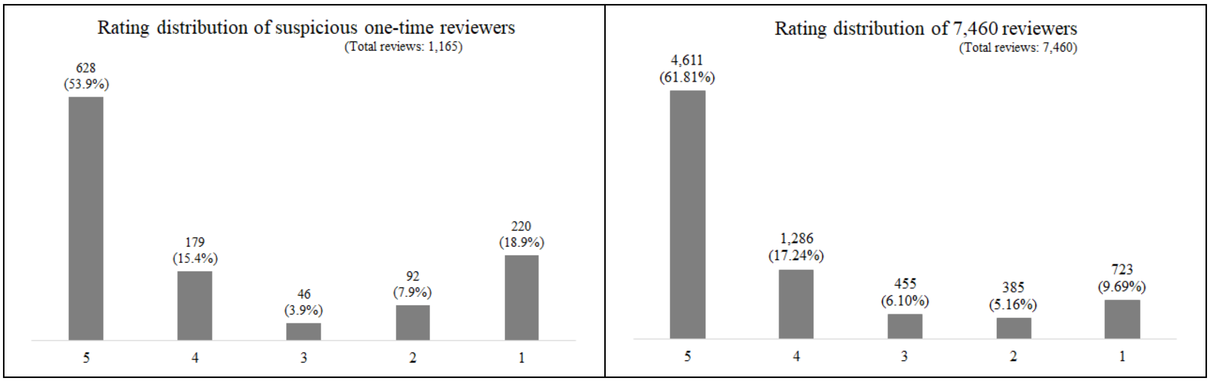


FIGURE 3. Rating distributions of Suspicious 1-time reviewers and reviewers after cleaning.

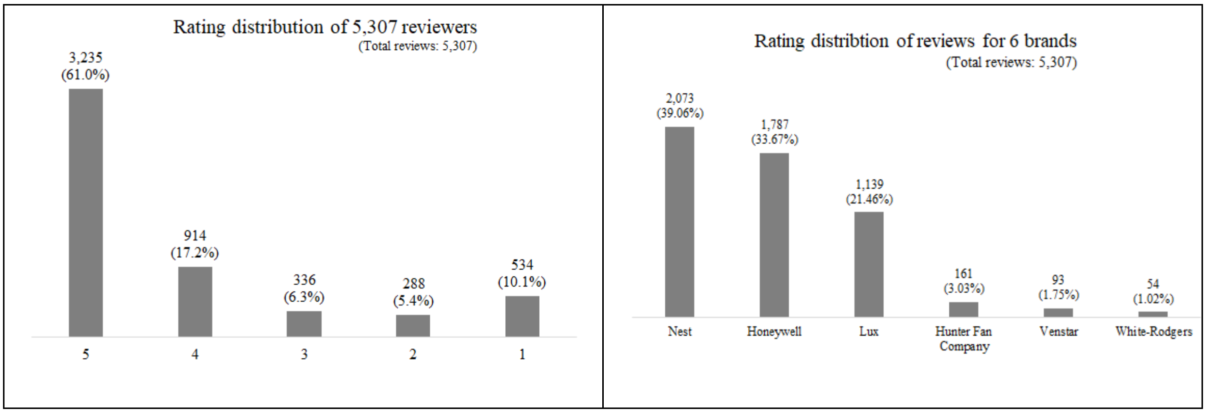


FIGURE 4. Rating distributions of reviews for six major brands.

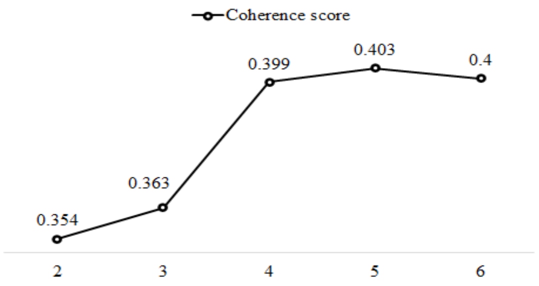


FIGURE 5. Coherence score over topic numbers.

Step 6: Modifying the PCDs by leveraging the domain expert's knowledge

The expert extends the five product content dimensions from the LDA model to nine dimensions based on domain knowledge and the purpose of the research design. The dimensions are: (1) smart connectivity, (2) easiness, (3) energy saving, (4) functionality, (5) support, (6) price value, (7) privacy, (8) the Amazon effect, and (9) environmental friendliness.

Passonneau *et al.* [23] suggested that annotation by experts transfers domain knowledge to machines for better prediction performance. Accordingly, the author manually annotates 47,763 labeling tasks for the reviewers' sentiment toward each product content dimension to transfer domain knowledge to the models (Table 15).

Dimension 1. Smart Connectivity

This dimension indicates the reviewers' sentiment toward programmable thermostats' (PTs') remote control of other home appliances through a Wi-Fi connection using apps and software. Wireless connectivity is a key component of thermostats' smartness as an Internet of Things (IoT) device because it enables consumers to control their home appliances with smartphones, tablets, and computers wherever and whenever they want.

Features related to remote control, Wi-Fi accessibility, and software quality for wireless control belong to this dimension. Firmware for Wi-Fi thermostats can update itself periodically and display customized pictures on the touch screen. For example, reviewers present positive sentiments like the following: "It is nice to monitor & adjust home temperature remotely on iPhone." and "I love the automatic updates that I have been receiving."

Dimension 2. Easiness

This dimension indicates the reviewers' sentiment toward PTs' simplicity and convenience of installation, set up,

TABLE 15. Topics in reviews after LDA and domain expert's annotation.

Topic dimensions	Interpretation
1. Smart-connectivity	The review describes WiFi, wireless connection issues with software (e.g., App) and hardware (e.g., Heating, ventilation, and air conditioning).
2. Easiness	The review mentions ease of use, including simplicity of installation, programming, and use.
3. Energy Saving	The review talks about energy savings, including money savings by reducing energy consumption.
4. Functionality	The review contains content related to the quality of the setting, controlling, and information related to temperature, time, scheduling, heating, and other devices.
5. Support	The review focuses on consumer support services before, during, and after they make a purchase.
6. Price value	The reviews discuss a reviewer's subjective evaluation about the price level compared with the quality, future benefits, and other factors.
7. Privacy	The reviews indicate privacy concerns related to thermostats.
8. Amazon Effect	The reviews mention Amazon's service quality, such as Amazon's delivery, consumer support, and refund and replacement policy.
9.Environmental friendliness	The reviews point out the issues related to carbon emissions and climate change.

programming, and usage. Unlike other experience goods, PTs require technical knowledge and skills. A lack of the required knowledge and skills may become a source of difficulty and failure of usage. The easiness of understanding the instruction manual, making the wiring connections, and controlling the device (including programming with a better user interface) belong to this dimension. Some reviewers posted ''Easy to Install and Use'' and ''so easy to use and so easy to see in the dark.''

Dimension 3. Energy Saving

This dimension indicates the reviewers' sentiment toward programmable thermostats' actual or expected energy saving and/or money saving due to better energy efficiency and cost-effectiveness than other thermostats or their previous one. The reviewers' comments about features related to better energy saving belong to this dimension along with the reduction of utility bills for electricity or gas. For example, reviews in this dimension include ''A much lower price in your electric bill.'' and ''My gas bill dropped by 30% the first month.''

Dimension 4. Functionality

The purpose of thermostats is to control energy usage for heating and cooling. Accurate and precise control for temperature and time are therefore essential for a better programmable thermostat. This dimension presents the quality of controlling and performance of features. The discomfort caused by thermostats' quality of functionality belongs to this dimension. For example, a clicking noise from thermostats during setting or programming indicates reviewers' negative sentiment toward this dimension. The reviews in this dimension include ''Temperature not accurate but does the job.'' and ''Makes a clicking noise.''

Dimension 5. Support

This dimension is related to consumer and technical support service, replacement and return service, warranty, packing quality, additional support service on the website, and other helpful materials for consumers. Consumer support services are vital for consumer satisfaction because thermostats require technical knowledge and skill during installation, setting up, and programming.

Consumer support services are vital for consumer satisfaction because thermostats require technical knowledge and skill during installation, setting up, and programming. Consumer support services may also mitigate inexperienced consumers' concerns, technical difficulties, and dissatisfaction during the pre- and post-purchase periods. Some reviews in this dimension are ''customer service is amazing! Tweet them for help even!'' and ''They sent mine in 2 days in perfect condition, plus they appear to have a fair return policy.''

However, the expert disregards the reviewers' sentiment toward Amazon's quality of consumer support service. Without separately considering the online market platform's service quality, the reviewers' sentiment toward this dimension for the PTs will be biased.

Dimension 6. Price Value

This dimension is a reviewer's subjective evaluation about the price level compared with the quality, future benefits, and other factors. Written comments related to the price value, all positive or negative events affecting the price, and repair costs belong to this dimension.

The prices on Amazon.com change very often and differ for consumers due to different promotions and memberships. The true price of reviewed products in the past may be different from the price at the time of web scraping. In this case, the observed price variables at the time of web scraping could be biased. Therefore, this study extracts the reviewers' sentiment toward this dimension from review text data. Some example reviews for this dimension are ''this is money well spent.'', ''Gold box deal makes it worth'', ''Too expensive to justify the benefit'', and ''running a promo to give you a $40 gift card with your purchase.''

Dimension 7. Privacy

This dimension is about privacy concerns related to thermostats. Wi-Fi thermostats provide remote control through the Internet, which may cause consumers to have concerns about privacy and data security. Wi-Fi thermostats can store and transform user information and consumption data.

Most of the negative privacy concerns occurred for the Nest when Google purchased it on January 13, 2014. Some reviews are ''Since Google's Nest buyout raises privacy concerns'' and ''Unless and until clear, unequivocal, irrevocable legal guarantees are in place that Google doesn't get Nest data, I would say that any Nest user must expect that, ultimately, Google will have all that data."

Dimension 8. The Amazon Effect

This dimension is the reviewers' sentiment caused by Amazon's service quality, such as Amazon's delivery, consumer support, and refund and replacement policy. Reviews on Amazon.com describe not only the product quality but also Amazon's service quality. If researchers do not account for the effect of Amazon's service quality on the reviewers' ratings, it may cause a bias. To the best of the author's knowledge, this is the first paper to measure the effect of Amazon's service quality on reviewers' star ratings.

Some reviews for this dimension are "Amazon's return policy is great!", "I am very pleased with this purchase and with Amazon customer service.", "Amazon is really good about their customer service", and "super fast Amazon delivery for free (overnight)."

Dimension 9. Environmental Friendliness

Since programmable thermostats are a home energy control device requiring energy consumption for heating and cooling, some researchers may be interested in the issues related to carbon emissions and climate change.

This dimension is a binary variable indicating whether reviews contain comments about the environmental friendliness of thermostats. Only nine reviews contain comments related to this dimension, including "it helps save the environment!", "I feel all environmentally friendly for wasting less energy, too.", and "thanks to this environmentally friendly thermostat. I am also helping to save the world."

APPENDIX C
HETEROSKEDASTICITY ORDERED PROBIT MODEL

Reviewers' observable ratings indicate the range of their unobservable continuous preference as follows:

$$\begin{aligned}
R_{ipt} &= 1, \quad \text{if } -\infty < U^*_{ipt} \le c_1 \\
R_{ipt} &= 2, \quad \text{if } c_1 < U^*_{ipt} \le c_2, \\
R_{ipt} &= 3, \quad \text{if } c_2 < U^*_{ipt} \le c_3, \\
R_{ipt} &= 4, \quad \text{if } c_3 < U^*_{ipt} \le c_4, \\
R_{ipt} &= 5, \quad \text{if } c_4 < U^*_{ipt} < \infty.
\end{aligned}$$

The ordered dependent variable, $R_{ipt} \in [1, 5]$, is reviewer i's first star rating for a PT on day t. U^*_{ipt} denotes the unobservable continuous utility of reviewer i for product p on day t. The unknown cutting points (thresholds) are denoted as c_k with the assumption that $c_1 < c_2 < c_3 < c_4$. U^*_{ipt} can be represented as follows:

$$U^*_{ipt} = x'_{ipt}\beta + \rho\varepsilon_{it}, \quad \varepsilon_{it} \sim \text{i.i.d Normal } (0, 1)$$

where x_{it} indicates a vector of independent variables, ε_{it} is a homoskedastic error term following a standard normal distribution, and $\rho > 0$ is a scale function to adjust the variance. The heteroskedasticity ordered probit (HETOP) model assumes its scaling function to be $\rho_i = \exp(Z'_{it}\gamma)$, where Z_i denotes the regressors for the scaling function and γ are unknown coefficients for Z_{it}. The probability of a reviewer's rating for a PT can be derived as follows:

$$\begin{aligned}
P(R_{ipt} = 1|x_{it}) &= P(\infty < U^*_{ipt} \le c_1|x_{it}) = \Phi\left(\frac{c_1 - x_{it}\beta}{\rho_i}\right) \\
P(R_{ipt} = 2|x_{it}) &= P(c_1 < U^*_{ipt} \le c_2|x_{it}) \\
&= \Phi\left(\frac{c_2 - x_{it}\beta}{\rho_i}\right) - \Phi\left(\frac{c_1 - x_{it}\beta}{\rho_i}\right) \\
P(R_{ipt} = 3|x_{it}) &= P(c_2 < U^*_{ipt} \le c_3|x_{it}) \\
&= \Phi\left(\frac{c_3 - x_{it}\beta}{\rho_i}\right) - \Phi\left(\frac{c_2 - x_{it}\beta}{\rho_i}\right) \\
P(R_{ipt} = 4|x_{it}) &= P(c_4 < U^*_{ipt} \le c_3|x_{it}) \\
&= \Phi\left(\frac{c_4 - x_{it}\beta}{\rho_i}\right) - \Phi\left(\frac{c_3 - x_{it}\beta}{\rho_i}\right) \\
P(R_{ipt} = 5|x_{it}) &= P(c_4 < U^*_{ipt} \le \infty|x_{it}) = 1 - \Phi\left(\frac{c_4 - x_{it}\beta}{\rho_i}\right)
\end{aligned}$$

where Φ is the cumulative distribution function (CDF) of the standard normal distribution. The log-likelihood (LL) function for N reviewers and reviews is:

$$\ln LL\,(\theta) = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{5} I\left(R_{ipt} = j\right) \ln P\left(R_{ipt} = j \mid x_i\right)$$

This LL function is maximized with respect to unknown parameters $\theta = \{\beta, \gamma, c_1, c_2, c_3, c_4\}$. $I(\cdot)$ denotes an indicator function and θ can be estimated through the maximum likelihood estimation.

Marginal effect analysis is an appropriate way to interpret each parameter in OP models. The variables in x_{it} can overlap with those in Z_{it}; therefore, x^a_{it} denotes the variables involved in both x_{it} and Z_{it} while x^b_{it} denotes the variables that only belong to x_{it}. In the case of continuous variables, Table 16 shows the marginal effects of both x^a_{it} and x^b_{it}.

The sign of a coefficient reflects the sign of the marginal effect only in the marginal effect of x^a_{it} at $R_{ipt} = 5$ and inversely reflects the sign of the marginal effect only in the marginal effect of x^a_{it} at $R_{ipt} = 1$. In all other cases, the sign of coefficient does not necessarily determine the sign of the marginal effect for the parameter. The marginal effect of the binary dummy at each level of $R_{ipt} = j \in [1, 5]$ can be derived as follows [59]:

$$\begin{aligned}
&\Delta P\left(R_{ipt} = j \mid x\right) \\
&\quad = P\left(R_{ipt} = j \mid x_{it}, d_{it} = 1\right) - P(R_{ipt} = j|x_{it}, d_{it} = 0)
\end{aligned}$$

where d_{it} is a binary dummy variable and $d_{it} = 0$ indicates the base group.

APPENDIX D
VARIABLES DESCRIPTIONS

See Table 17.

The category diversity is the Shannon index as follows:

$$\text{Diverisity index}_{i,t_i} = -\sum\nolimits_{c=1}^{C} P_{c,t_i} \ln P_{c,t_i},$$

TABLE 16. The marginal effect of the HETOP model.

	(1) case of $x_{it}^{a} \in x_{it} \cap Z_{it}{}^{c}$	(1) case of $x_{it}^{b} \in x_{it} \cap Z_{it}$
The marginal effect of x_{it} at R_{ipt}= 1	$\emptyset\left(\frac{c_1 - x'_{it}\beta}{\exp(Z'_{it}\gamma)}\right)\frac{-\beta_{x_{it}^{a}}}{\exp(Z'_{it}\gamma)}$	$\emptyset\left(\frac{c_1 - x'_{it}\beta}{\exp(Z'_{it}\gamma)}\right)\left(\frac{-\beta_{x_{it}^{b}} - (c_1 - x'_{it}\beta)\gamma_{x_{it}^{b}}}{\exp(Z'_{it}\gamma)}\right)$
The marginal effect of x_{it} at R_{ipt}= j where j ∈ {2, 3, 4}	$\left[\emptyset\left(\frac{c_j - x'_{it}\beta}{\exp(Z'_{it}\gamma)}\right) - \emptyset\left(\frac{c_{j-1} - x'_{it}\beta}{\exp(Z'_{it}\gamma)}\right)\right]\frac{-\beta_{x_{it}^{a}}}{\exp(Z'_{it}\gamma)}$	$\left[\emptyset\left(\frac{c_j - x'_{it}\beta}{\exp(Z'_{it}\gamma)}\right)\left(\frac{-\beta_{x_{it}^{b}} - \left(c_j - x'_{it}\beta\right)\gamma_{x_{it}^{b}}}{\exp(Z'_{it}\gamma)}\right)\right]$ $-\left[\emptyset\left(\frac{c_{j-1} - x'_{it}\beta}{\exp(Z'_{it}\gamma)}\right)\left(\frac{-\beta_{x_{it}^{b}} - (c_{j-1} - x'_{it}\beta)\gamma_{x_{it}^{b}}}{\exp(Z'_{it}\gamma)}\right)\right]$
The marginal effect of x_{it} at R_{ipt}= 5	$\emptyset\left(\frac{c_4 - x'_{it}\beta}{\exp(Z'_{it}\gamma)}\right)\frac{\beta_{x_{it}^{a}}}{\exp(Z'_{it}\gamma)}$	$\emptyset\left(\frac{c_4 - x'_{it}\beta}{\exp(Z'_{it}\gamma)}\right)\left(\frac{\beta_{x_{it}^{b}} + (c_4 - x'_{it}\beta)\gamma_{x_{it}^{b}}}{\exp(Z'_{it}\gamma)}\right)$

* Notes: $\emptyset(\cdot)$ indicates the probability density function (PDF) of the standard normal distribution

where $P_{c,t_i} = \frac{N_{c,t_i^b}}{\sum_{c=1}^{C} N_{c,t_i^b}}$ and N_c is the number of prior reviews in subcategory c by t_i^b.

APPENDIX E MARGINAL EFFECT

Tables in this section show the marginal effect of key variables (model_h2) at the average value of one company's reviewers (Nest, during June 2014).

APPENDIX F MACHINE LEARNING MODELS

Six popular machine learning models are applied to ex ante prediction tasks. The support vector machine and decision tree models are base models used to compare their prediction performance with more complex models. Random forest and extreme gradient boosting are tree ensemble models. The artificial neural net and long–short-term memory models are deep learning models. A high-level overview of each model is presented below.

A. KERNEL SUPPORT VECTOR MACHINE (KERNEL SVM)

The support vector machine (SVM) model finds the linear separable hyperplane in the feature space to classify labels [38]. To deal with non-linearly separable, noisy, and outlier data, Cortes and Vapnik [60] introduced a slack variable as $\xi_i \geq 0, \forall i$ and a parameter C. ξ_i is the distance between the linear hyperplane and the misclassified x_i, while C is a weight for the sum of ξ_i in the sample as $\sum_{i=1}^{N} \xi_i$ [61].

In particular, kernel SVM is applied in this study to consider the non-linearity of the data. A kernel function K implicitly maps original data to a high-dimensional functional feature space $\Phi : x \rightarrow \varphi(x)$, such that $K\left(x,x'\right) =< \varphi(x), \varphi(x') >$ for two samples x and x'. The Gaussian radial basis function (RBF) is the kernel function, as follows:

$$K_{rbf}(x, x') = \exp\left(-\gamma\,||x - x'||_2^2\right))$$

where $\gamma > 0$ and $||x - x'||^2$ is the squared Euclidean distance between x and x'. The RBF is a similarity measure ranging between zero and one, and $\varphi(x)$ has an infinite number of dimensions [62].

Overall, the dual problem of kernel SVM can be expressed as follows:

$$\max_{\alpha_i} \sum_{i=1}^{N} \xi_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i\alpha_j y_i y_j K_{rbf}(x_i, x_j)$$

where $C \geq \alpha_i \geq 0$ and $\sum_{i=1}^{N} \alpha_i y_i = 0$. α_i denotes the Lagrange multipliers, and $\{x_i|C > \alpha_i > 0, \forall i\}$ are the support vectors deciding the decision boundary. C is an upper bound of ξ_i in this kernel SVM optimization setting. In addition, C and γ are two hyperparameters of SVM.

One-vs-rest (OvR) is a popular method for multiclass classification [63]. In the OvR approach to three-class classification, three binary SVMs classify each class in an online product review against the rest of the classes as {1, the others}, {2, the others}, and {3, the others}. The SVM that has the largest margins among the three SVMs determines the class of new data in the test set.

B. DECISION TREE (DT)

The decision tree (DT) model recursively partitions the feature space into a disjointed set of rectangular regions such that each region contains the same classes (Figure 7). For multiclass classification, the DT model has K classes (K > 2). The feature space at each node n is divided into two sub-regions based on $\theta_n \in \{x_j, t_j|node = n\}$, where x_j denotes the splitting variable j and t_j denotes the splitting value for x_j at node n. θ_n splits the data at node n into $\{D_{left}(\theta_n)|x_j \leq t_j\ at\ node = n\}$ and $\{D_{right}(\theta_n)|x_j > t_j\ at\ node = n\}$. R_n represents the region corresponding to node n in the feature space, and $N_n = \sum_{i=1}^{N} I(x_i \in R_n)$ means the total number of instances in R_n. Node m denotes the terminal node. The hyperparameter of DT is the maximum number of the tree depth in this study.

In DT, impurity means the heterogeneity of classes in a node and H (·) denotes the impurity function. The optimal value of θ_n^* minimizes the impurity at the given node n as

TABLE 17. Variables generated from user and crowd DFs (N= 5,307).

Variable	Description	Mean	SD	Min	Max
rating (dependent)	i (the reviewer)'s five-scale star-rating for a PT at t_i*	4.136	1.33	1	5
sum_len	i's length of review summary (headline) at t_i	28.62	17.78	2	134
rev_len	i's length of review body at t_i	796.84	1,007.17	0	11,981
title_len	The length of tittle for the PT reviewed by i at t_i	55.67	10.20	31	107
desc_len	The length of description for the PT reviewed by i at t_i	1,298.43	1,526.47	0	4,788
Nest	Brand dummy for the Nest (base group is White Roger)	0.39	0.49	0	1
honey	Brand dummy for the Honeywell	0.34	0.47	0	1
hunter	Brand dummy for the Hunter Fan	0.03	0.17	0	1
Lux	Brand dummy for the Lux	0.21	0.41	0	1
venstar	Brand dummy for the Venstar	0.02	0.13	0	1
Price	p (the PT reviewed by i at t_i)'s price (at the time of web scrapping)	156.53	114.78	14.99	350.3
u_avg_p_dfs	i's average price for reviewed products in all categories by t_i^b*	62.31	70.81	0	899.99
u_sd_p_dfs	i's SD of price for reviewed products in all categories by t_i^b	55.15	67.63	0	629.32
u_max_p_dfs	i's the highest price among reviewed items in all categories by t_i^b	194.50	212.68	0	999.99
u_help_dfs	The number of helpfulness upvote for i in all categories by t_i^b	4.14	15.70	0	911
u_no_help_dfs	The number of helpfulness downvote for i in all categories by t_i^b	1.03	3.49	0	170
u_avg_len_sum	i's average length of summary in all categories by t_i^b	25.35	11.12	1	125
u_sd_len_sum	i's SD of length of summary in all categories by t_i^b	8.76	7.09	0	64.35
u_avg_len_rev	i's average length of review body in all categories by t_i^b	558.21	504.42	71	7,354
u_sd_len_rev	i's SD of length of review body in all categories by t_i^b	305.07	435.72	0	8,139.37
sum_amz_video	i's number of reviews in the amazon instant video category by t_i^b	0.00	0.08	0	3
sum_appliance	i's number of reviews in the appliance category by t_i^b	0.05	0.26	0	4
sum_apps	i's number of reviews in the apps for android category by t_i^b	0.00	0.03	0	1
sum_arts_crafts	i's number of reviews in the art crafts category by t_i^b	0.06	0.54	0	30
sum_automotive	i's number of reviews in the automotive category by t_i^b	0.48	1.69	0	35
sum_baby	i's number of reviews in the baby category by t_i^b	0.18	1.14	0	28
sum_beauty	i's number of reviews in the beauty category by t_i^b	0.24	1.90	0	93
sum_books	i's number of reviews in the book category by t_i^b	2.46	17.99	0	857
sum_buyakindle	i's number of reviews in the kindle category by t_i^b	0.02	0.21	0	8
sum_cdsvinyl	i's number of reviews in the cds and vinyl category by t_i^b	0.33	2.06	0	92
sum_cellphone	i's number of reviews in the cell phones category by t_i^b	0.62	2.04	0	55
sum_clothes	i's number of reviews in the clothes, shoes, jewelry category by t_i^b	0.25	1.01	0	44
sum_computers	i's number of reviews in the computer category by t_i^b	0.00	0.08	0	2
sum_digit_music	i's number of reviews in the digital music category by t_i^b	0.03	0.31	0	11
sum_electronics	i's number of reviews in the electronics category by t_i^b	3.20	9.88	0	386
sum_giftcards	i's number of reviews in the gift cards category by t_i^b	0.00	0.08	0	2
sum_grocery	i's number of reviews in the grocery gourmet food category by t_i^b	0.38	3.90	0	218
sum_healthcare	i's number of reviews in the health personal care category by t_i^b	0.734	3.88	0	196
sum_home_kitch	i's number of reviews in the home kitchen category by t_i^b	1.08	3.72	0	137
sum_industry_spe	i's number of reviews in the industry specific category by t_i^b	0.10	0.57	0	23
sum_kindle_store	i's number of reviews in the kindle store category by t_i^b	0.00	0.06	0	3
sum_magazine	i's number of reviews in the magazine subscription category by t_i^b	0.01	0.19	0	10
sum_movies_tv	i's number of reviews in the move and tv category by t_i^b	0.78	7.13	0	199
sum_musical_ins	i's number of reviews in the musical instrument category by t_i^b	0.11	1.10	0	58
sum_office_prod	i's number of reviews in the office products category by t_i^b	0.53	2.66	0	127
sum_patio_lawn	i's number of reviews in the patio, lawn, and garden category by t_i^b	0.36	1.37	0	33
sum_pet_supp	i's number of reviews in the pet supplies category by t_i^b	0.26	1.38	0	39
sum_software	i's number of reviews in the software category by t_i^b	0.17	1.13	0	42
sum_sports_out	i's number of reviews in the spots and outdoors category by t_i^b	0.57	3.99	0	260
sum_tools_home	i's number of reviews in the tools & home category by t_i^b	1.13	3.88	0	109
sum_toys_games	i's number of reviews in the tops and games category by t_i^b	0.30	1.84	0	67
sum_video_games	i's number of reviews in the video games category by t_i^b	0.32	2.26	0	66
u_cum_reviews	i's number of reviews in all categories by t_i^b	14.81	53.39	1	2,429
u_cate_diversity	Shanon index for i's category diversity of reviews posted by t_i^b	0.98	0.74	0	2.74
u_avg_rating	i's average star-rating in all categories by t_i^b	3.98	0.99	1	5
u_sd_rating	i's SD of star-rating in all categories by t_i^b	0.83	0.72	0	2.83
c_cum_reviews	p's number of crowd's reviews by t_i^b	524.74	639.35	1	2,425
c_avg_rating	p's average rating of crowd by $t_{j\neq i}^b$*	4.20	0.31	1	5
c_sd_rating	p's SD of crowd's rating by $t_{j\neq i}^b$	1.23	0.30	0	2.83
c_avg_len_sum	p's average length of review summary written by crowd until $t_{j\neq i}^b$	27.55	2.99	4	55
c_sd_len_sum	p's SD of review summary written by crowd until $t_{j\neq i}^b$	16.24	3.07	0	36.89

TABLE 17. ***(Continued.)*** **Variables generated from user and crowd DFs (N= 5,307).**

c_avg_len_rev	p's average length of review body written by crowd until $t^b_{j\neq i}$	826.66	334.08	103	4,384.67
c_sd_len_rev	p's SD for the length of review body written by crowd until $t^b_{j\neq i}$	951.83	489.97	0	5,676.47
c_rating_rec	p's average rating of crowd at $t^b_{j\neq i}$	4.13	1.34	1	5
c_len_sum_rec	p's the length of review summary written by a crowd at $t^b_{j\neq i}$	27.31	17.19	1	134
c_len_rev_rec	p's the length of review body written by a crowd at $t^b_{j\neq i}$	704.78	920.48	0	11,981
Day	Day dummies for t_i and base day is Monday (0)	2.88	1.98	0	6
month	Month dummies for t_i and base month is January (1)	2,012.45	1.50	2005	2014
Year	Year dummies for t_i and base year is 2005	6 .35	3.87	1	12
holiday	US holiday dummies and base is not holiday (0)	0.03	0.17	0	1
interval	The time interval between p's the day of first review and t_i	990.38	841.96	1	3,399
nest_avail	Dummy for the first day of the Nest's PT on Amazon (Dec 15, 2011)	0.82	0.38	0	1
smart_con	i's sentiment of p's smart connectivity in i's review at t_i	0.19	0.49	-1	1
Easy	i's sentiment of p's easiness in i's review at t_i	0.41	0.67	-1	1
Save	i's sentiment of p's energy saving in i's review at t_i	0.18	0.43	-1	1
Func	i's sentiment of p's functionality in i's review at t_i	0.16	0.80	-1	1
support	i's sentiment of p's support in i's review at t_i	-0.00	0.39	-1	1
price value	i's sentiment of p's perceived price value in i's review at t_i	0.10	0.48	-1	1
privacy	i's sentiment of p's privacy issues in i's review at t_i	-0.00	0.07	-1	1
amazon	i's sentiment of p's Amazon effect in i's review at t_i	0.01	0.16	-1	1
Env	i's sentiment of p's environmental friendliness in i's review at t_i	0.00	0.04	0	1

* Notes : t_i = the day when reviewer i wrote a review about a PT (p) for the first time; ** $t^b_i = \underset{t^b_i < t_i}{\operatorname{argmin}} |t_i - t^b_i|$, the most recent day when reviewer i wrote a review before t_i; $t^b_{j\neq i} = \underset{t^b_j < t_i}{\operatorname{argmin}} |t_i - t^b_j|$, the most recent day when the reviewer j wrote a review before t_i; and symbol u in front of the variables (e.g., u_avg_rating) indicates user DFs while c indicates crowd DFs (e.g., c_avg_rating).

follows:

$$\theta^*_n = \underset{\theta_n}{\text{agmin}} \frac{[N_{left|n} H(\{D_{left}(\theta_n)) + N_{right|n} H(\{D_{right}(\theta_n))]}{N_n}$$

where $N_n = N_{left|n} + N_{right|n}$.

Entropy is the impurity measure in this study and can be expressed as follows:

$$H\,(D(\theta_n)) = -\sum_{k=1}^{K} p_{kn}(1 - p_{kn})$$

where $p_{kn} = \frac{1}{N_n} \sum_{\substack{k=1 \\ x_i \in R_n}}^{K} I(y_i = k)$.

The decision tree is simple, interpretable, applicable for regression and classification with continuous and/or categorical variables, and acceptable for a dataset containing missing values. However, the decision tree has high variance due to its hierarchical structure, which means that a small change in features can cause different split results. Further, the classification of the DT on imbalanced data could be biased toward the majority class. Therefore, tree ensemble models are applied to mitigate these problems.

C. RANDOM FOREST (RF)

Ensemble methods use a set of base classifiers. The random forest (RF) is a tree ensemble model called bootstrap aggregating. Dietterich (2000) suggested that ensemble models often perform better than single classifiers because (1) averaging classifiers may reduce the probability of using the wrong classifier; (2) different starting points for each classifier's optimization may reduce the possible local optima; and (3) combining classifiers may represent the correct function for mapping features to labels [40].

In particular, the RF is able not only to improve the prediction performance by reducing variation but also to maintain robust prediction performance with an increasing number of noisy variables [41].

The RF' procedure is: (1) generating an independent training set s_i by selecting a subset of the sample from training set S with replacement; (2) creating de-correlated RF rf_i, by selecting a subset of features; (3) training rf_i with s_i and using fitted rf_i to classify new data x; and (4) repeating the above steps B times and classifying new data by using majority voting as follows:

$$\hat{y} = \frac{1}{B} \sum_{i=1}^{B} rf_i(x; \theta_i)$$

where θ_i indicates the parameters determining the structure of rf_i, including the subset of features, splitting variables and points at each node, and the values at each terminal node. The hyperparameters are the number of trees and the depth of the trees.

Breiman [64] argued that the RF's prediction performance depends on the performance of individual DTs and the correlation between DTs. However, the minority classes in imbalanced data could be less represented in the sub-samples due to resampling, and this may cause lower prediction performance for the minority classes in RF. Chen *et al.* [36] suggested using the weighted RF to correct the problem of imbalance.

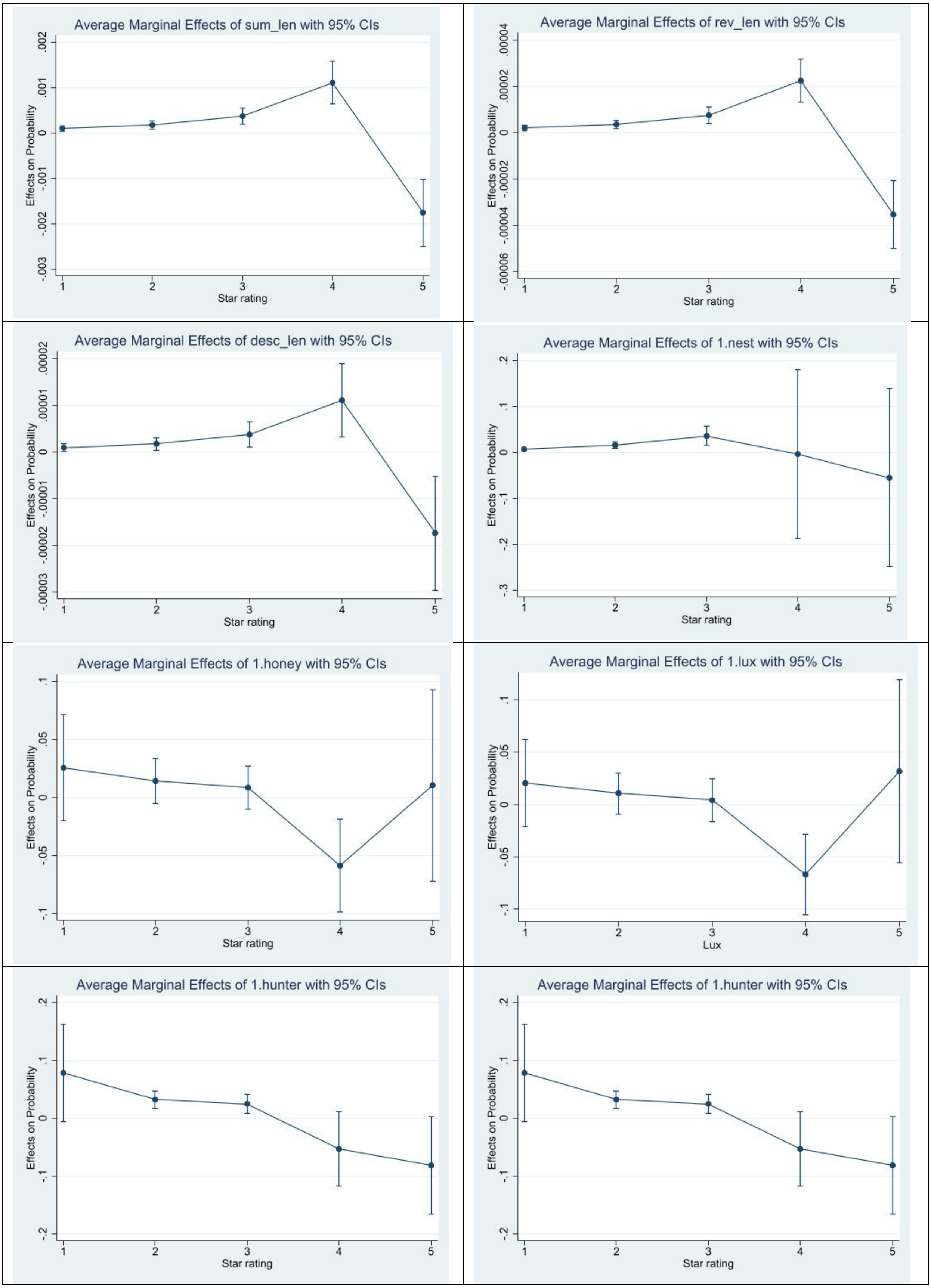
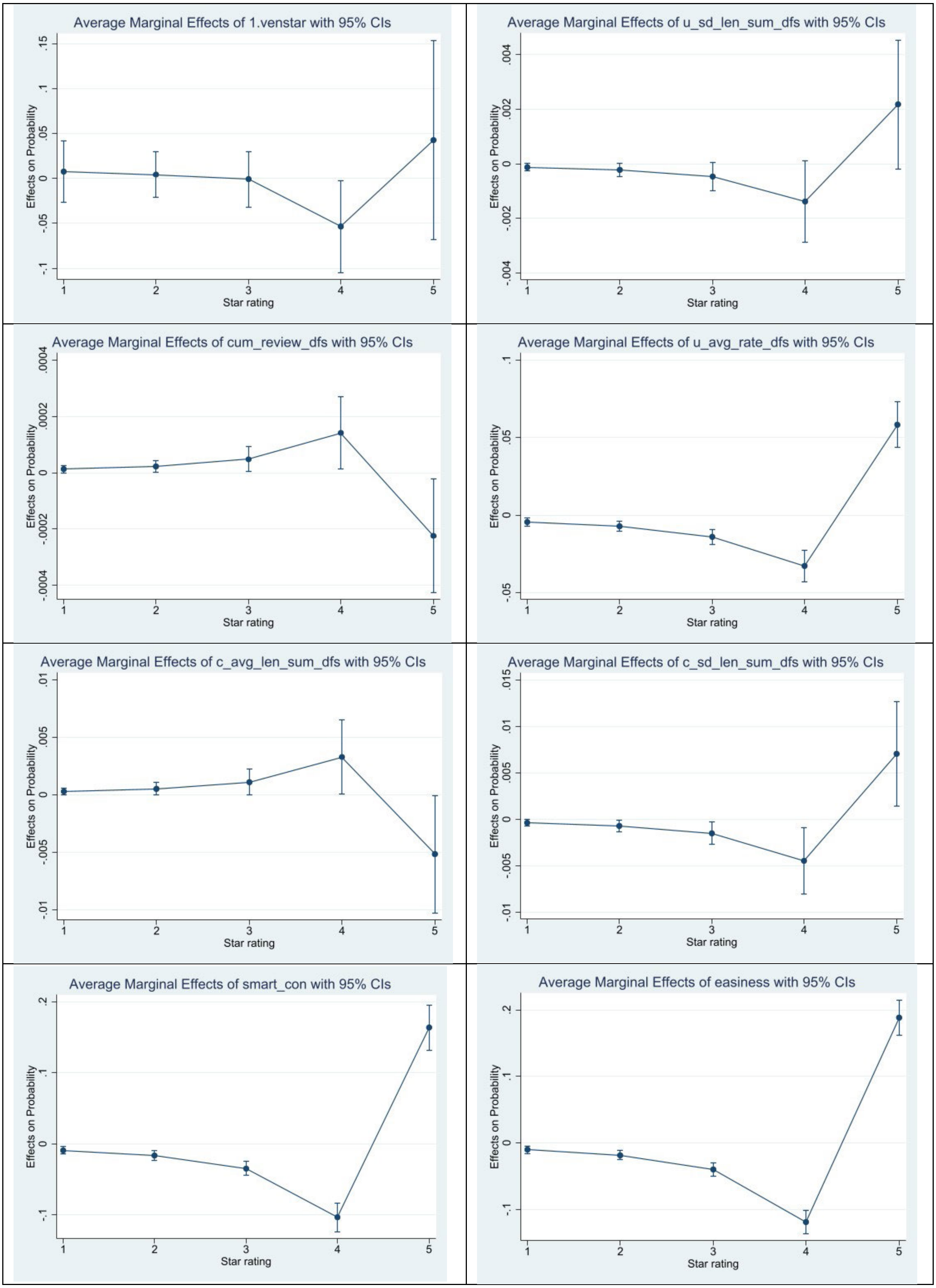


FIGURE 6. **Marginal effect (statistically significant variables or related variables).**



Average Marginal Effects of 1.venstar with 95% CIs

Average Marginal Effects of u_sd_len_sum_dfs with 95% CIs

Average Marginal Effects of cum_review_dfs with 95% CIs

Average Marginal Effects of u_avg_rate_dfs with 95% CIs

Average Marginal Effects of c_avg_len_sum_dfs with 95% CIs

Average Marginal Effects of c_sd_len_sum_dfs with 95% CIs

Average Marginal Effects of smart_con with 95% CIs

Average Marginal Effects of easiness with 95% CIs

FIGURE 6. ***(Continued.)*** **Marginal effect (statistically significant variables or related variables).**

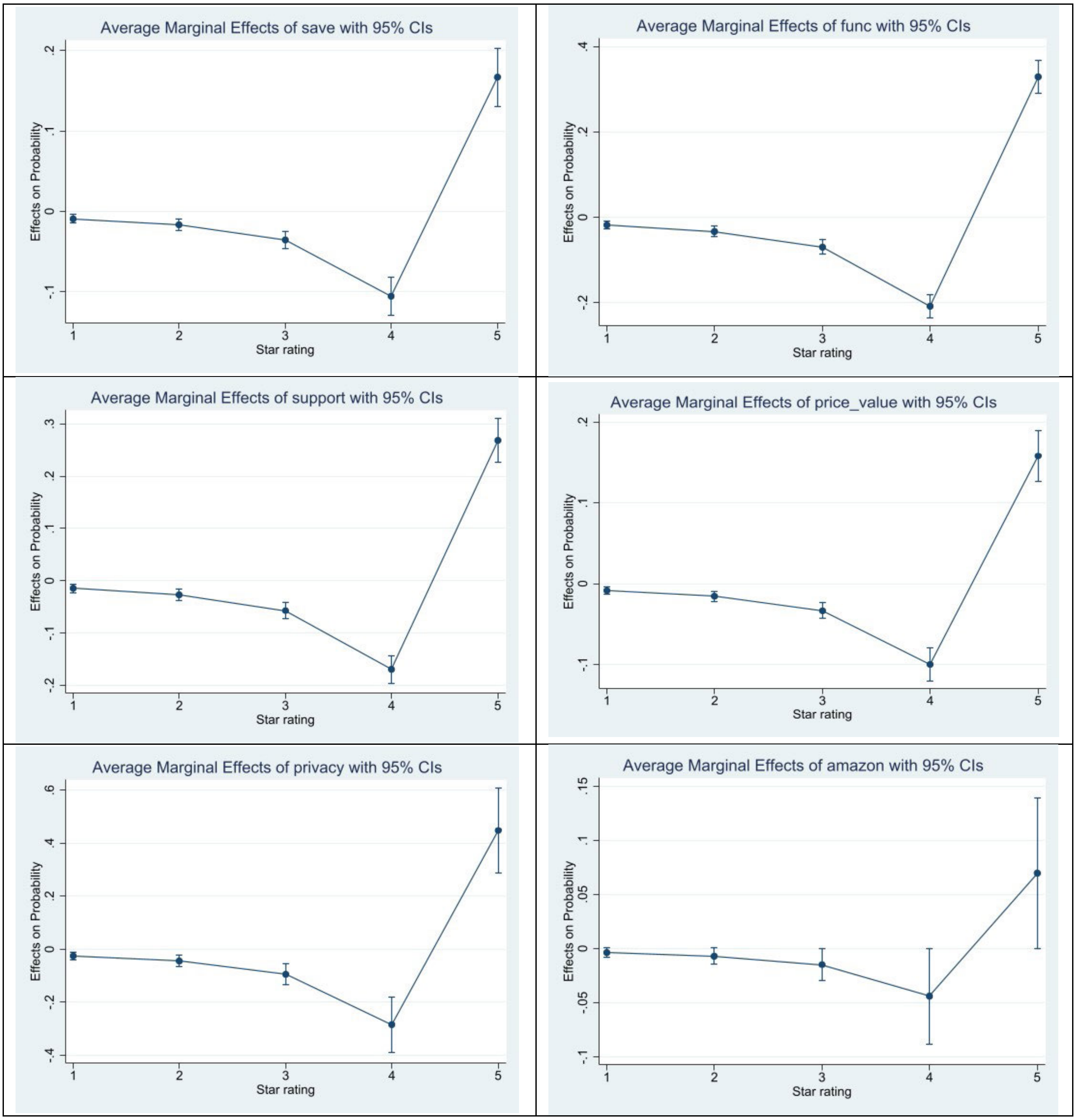


FIGURE 6. *(Continued.)* **Marginal effect (statistically significant variables or related variables).**

D. EXTREME GRADIENT BOOSTING (XGB)

Boosting combines multiple weak classifiers to build a strong classifier. However, boosting does not involve bootstrap resampling [39]. Extreme gradient boosting (XGB) [42] implements gradient boosting [65] by regularizing the complexity of the tree structure. The prediction of a tree ensemble model is the sum of K DTs:

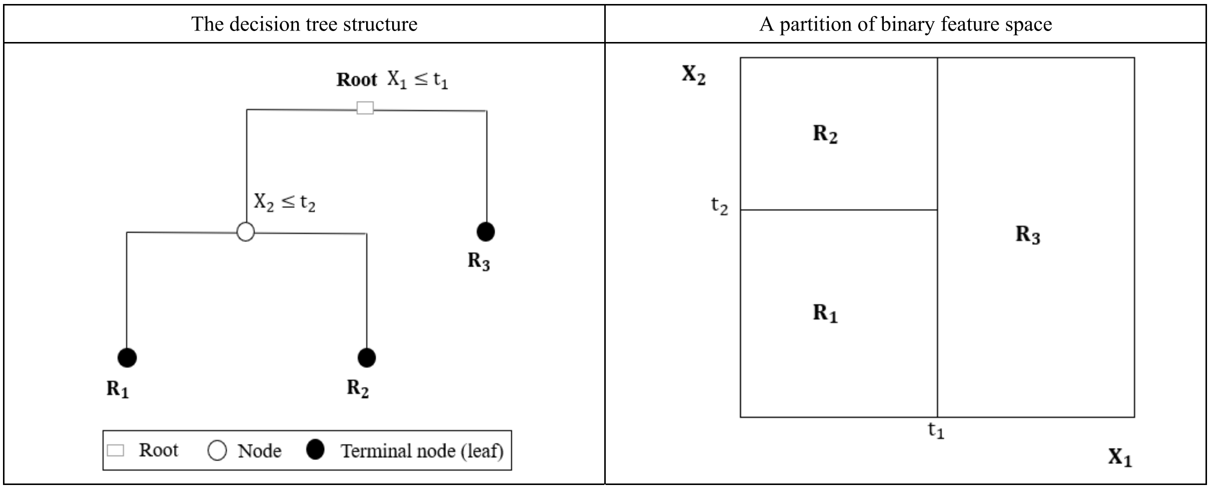
$$\widehat{y_i} = \sum_{k=1}^{K} f_k(x_i), \qquad f_k \in F$$

where $F = \{f(x) = w_{q(x)} | q(x) \in \{1, .., T\}$ and $w \in R^T\}$. F is a possible functional space of DTs, q is a leaf index function and represents the structure of the tree, T is the number of leaves in the tree, and w is the weight of each leaf.

Each DT has an objective function (OF). A smaller OF value means a better tree structure. The optimization of each tree structure minimizes the OF:

$$OF = \sum\nolimits_i^N L(y_i, \widehat{y}_i) + \sum\nolimits_{k=1}^K \left[\gamma T + \frac{1}{2}\lambda ||w||^2\right]$$

The OF contains additive tree functions; therefore, it cannot be optimized by the conventional methods. Therefore, additive training is applied to the optimization by adding a new function $f_t(x_i)$ in each iteration t and using a second-order


The decision tree structure

A partition of binary feature space

FIGURE 7. Decision tree structure.

Taylor approximation:

$$OF^{(t)} \approx \sum_{i}^{N} L(y_i, \widehat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) + \sum_{k=1}^{K} \left[\gamma T + \frac{1}{2}\lambda ||w||^2 \right]$$

where $g_i = \frac{\partial L(y_i, \widehat{y}_i^{(t-1)})}{\partial \widehat{y}_i^{(t-1)}}$ and $h_i = \frac{\partial^2 L(y_i, \widehat{y}_i^{(t-1)})}{\partial \widehat{y}_i^{(t-1)}}$.

For the multiclass classification, the softmax loss (cross entropy loss) is applied:

$$L(y_i, \widehat{y}_i) = -\alpha_k \sum_{k=1}^{K} I(y_i = k) \log \Pr(\widehat{y}_i = y_i | x)$$

For imbalanced data, α_k becomes $\frac{N}{K \times N_k}$ to put more weight on the minority class and less on the majority class in the loss function [66]. The hyperparameters of XGB in this study are the number of trees, tree depth, learning rate, and class weight.

E. ARTIFICIAL NEURAL NETWORK (ANN)

An ANN is a deep learning (DL) model. DL automatically learns a representation of data for required tasks [43]. Recently, deep learning has shown dramatic progress in diverse areas including natural language processing (NLP). Deep learning also has the potential to improve business analytics [67].

Deep learning relies on the universal approximation theorem [68], [69]. In this theorem, an ANN represented by $\hat{F}(x, w)$ can approximate any Borel measurable function f(x) (any continuous function on a compact subset of finite Euclidean space is Borel measurable) with any desired degree of accuracy [43], [70] as follows:

$$\text{If } \forall\, f(x) \text{ is continous in } R^n, \text{ there is weight vector } w \text{ in } |\hat{F}(x, w) - f(x)| < \varepsilon, \forall x$$

The ANN will also be useful for approximating E(Y|X) by mitigating functional form misspecification [44], [71].

The ANN has a multilayer structure with input, hidden, and output layers. Figure 8 shows the basic structure of the ANN for binary classification. The ANN example has an input layer with two input variables, one hidden layer with three neurons, and one output layer. Each neuron in the hidden layer receives a weighted input value from the input layer and the received input values enter the activation function (continuous nonlinear function) in each neuron. In this example, the activation function is the rectified linear unit (ReLU), $f(x) = \max(0, x)$. The weighted sum of output values from the hidden layer enters the output layers. The softmax function, $f(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_i)}$, turns the output values from the previous hidden layer into the probability of class one. If $P(\text{class} = 1) > 0.5$, the label will be one; otherwise, it will be zero. The ANN learns optimal weights by backpropagation [72].

In this study, the ANN structure contains two hidden layers. The activation functions are ReLU. The optimization method for minimizing cross-entropy loss is Adam [73]. Dropout is a regularization method used to prevent overfitting during the training steps.

The hyperparameters are the optimal training iteration, dropout rate, learning rate, and number of neurons in the two hidden layers. The class weight is also a hyperparameter; however, the class-weighted ANN shows lower prediction performance than the unweighted one.

F. LONG SHORT-TERM MEMORY (LSTM)

The recurrent neural net (RNN) is a DL model for sequence data. However, the RNN may suffer from the vanishing gradient problem during the training of long sequence data [74]. LSTM mitigates the vanishing gradient problem by introducing the memory cell structure [45], [75].

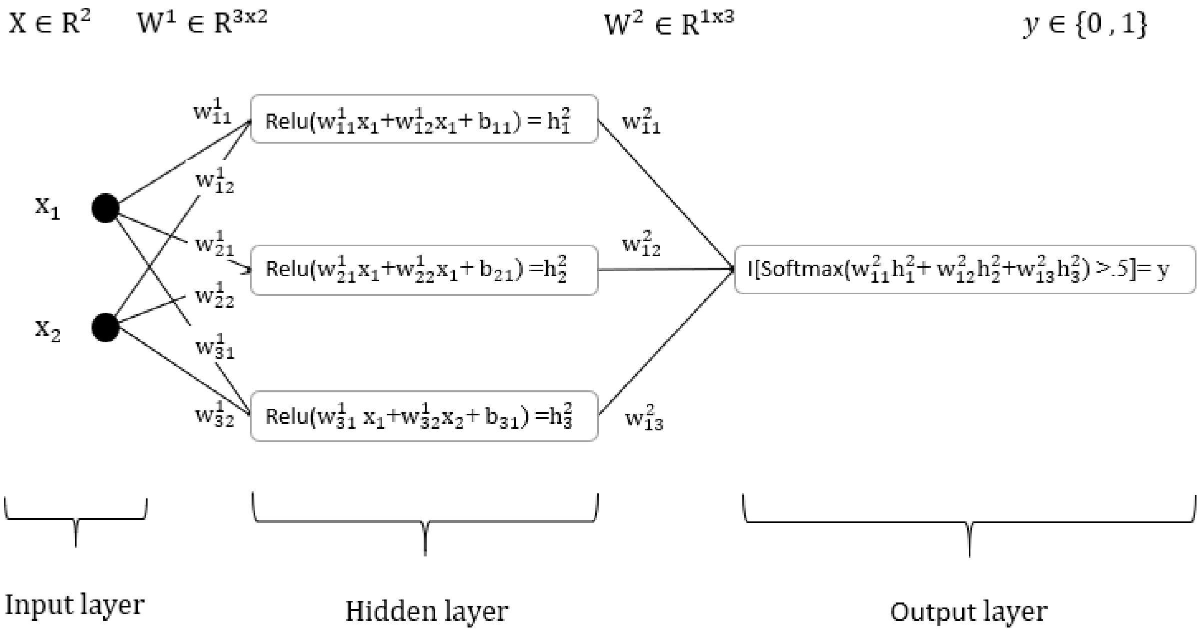

$X \in R^2$
$W^1 \in R^{3x2}$
$W^2 \in R^{1x3}$
$y \in \{0, 1\}$
x_1
x_2
w^1_{11}
w^1_{12}
w^1_{21}
w^1_{22}
w^1_{31}
w^1_{32}
$\text{Relu}(w^1_{11}x_1+w^1_{12}x_1+b_{11}) = h^2_1$
$\text{Relu}(w^1_{21}x_1+w^1_{22}x_1+b_{21}) = h^2_2$
$\text{Relu}(w^1_{31}x_1+w^1_{32}x_2+b_{31}) = h^2_3$
w^2_{11}
w^2_{12}
w^2_{13}
$\text{I}[\text{Softmax}(w^2_{11}h^2_1+w^2_{12}h^2_2+w^2_{13}h^2_3) > .5] = y$
Input layer
Hidden layer
Output layer

FIGURE 8. Example of the ANN structure.

LSTM has a multilayer structure with input, hidden, and output layers. In particular, the hidden layer(s) contains memory cells. Each memory cell is controlled by three gates (the input i_t, forget gate f_t, and output gate o_t). The memory cell at time t receives the input value x_t, hidden state h_{t-1} and previous cell state at t-1 C_{t-1}.

The input gate i_t decides whether the information in x_t and h_{t-1} is useful for C_t. The forget gate f_t decides whether the information in h_{t-1} is useful for C_t. The output o_t decides which information in C_t will be preserved in h_t. Figure 9 shows the structure of the memory cell. The hyperparameters of the LSTM model in this study are the learning rate, training epochs, and number of neurons.

APPENDIX G
EX ANTE PREDICTION RESULTS

Model 1 (''at time model'') is the base model that contains only 37 observable variables. Models 2, 3, and 4 are ex ante models used to predict consumers' potential sentiment for PTs before they make a purchase. Models 5 and 6 are ''partial ex ante'' models used to predict consumers' potential sentiment for the PTs purchased before they write a review.

APPENDIX H
WORD EMBEDDING METHODS

A. TERM FREQUENCY–INVERSE DOCUMENT FREQUENCY (TF-IDF)

Frequency-based embedding is a simple way to map each review text to numerical vectors. Term frequency–inverse document frequency (TF-IDF) is a frequency-based type of word embedding and penalizes the high-frequency words in the entire review [35]. For example, ''the'' may have a low TF-IDF value because many reviews contain ''the''.

The pre-processing for TF-IDF in this study is conducted as follows:

Step 1. Putting all words into lower case;

Step 2. Splitting the review text into words;

Step 3. Removing stopwords, punctuation, numbers, and single characters;

Step 4. Lemmatizing words (converting words into the base form, e.g., writing → write).

After the above steps, the number of unique words in 5,307 review texts (vocabulary) is 15,843. This is a spare high-dimension matrix containing many zero values. TF-IDF represents how frequently a word appears in the entire review as follows:

$$\text{TF}-\text{IDF score}(\text{unique word}_{n,i}) = \text{tf}_{n,i} \times \log\frac{N}{\text{df}_n}$$

$\text{tf}_{n,i}$: the frequency of word n in review i (term frequency)

df_n : the frequency of reviews containing word n (document frequency)

N : the number of total reviews ($N = 5{,}307$)

In this equation, low-frequency words in review i will have a low TF-IDF score due to low term frequency; common words that occur in many reviews will also have a low TF-IDF score due to low document frequency [78]. On top of the TF-IDF embedding vectors from the review text data, tree ensemble models (RF and XGB) are applied for sentiment analysis. TF-IDF has a high-dimensional spare matrix and cannot represent similarity, ambiguity, and contextual meaning in a text.

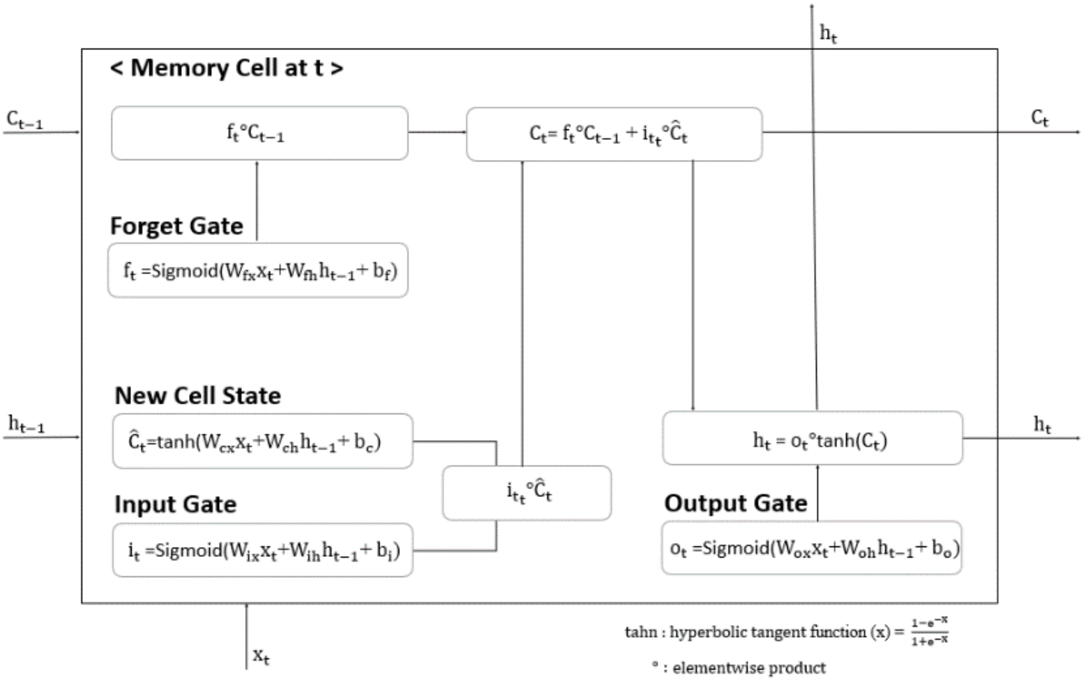

< Memory Cell at t >
C_{t-1}
$f_t \circ C_{t-1}$
$C_t = f_t \circ C_{t-1} + i_{t_t} \circ \hat{C}_t$
C_t
h_t
Forget Gate
f_t =Sigmoid($W_{fx}x_t$+$W_{fh}h_{t-1}$+ b_f)
New Cell State
h_{t-1}
$\hat{C}_t$=tanh($W_{cx}x_t$+$W_{ch}h_{t-1}$+ b_c)
$i_{t_t} \circ \hat{C}_t$
$h_t = o_t \circ \tanh(C_t)$
h_t
Input Gate
i_t =Sigmoid($W_{ix}x_t$+$W_{ih}h_{t-1}$+ b_i)
Output Gate
o_t =Sigmoid($W_{ox}x_t$+$W_{oh}h_{t-1}$+ b_o)
x_t
tahn : hyperbolic tangent function $(x) = \frac{1-e^{-x}}{1+e^{-x}}$
$\circ$: elementwise product

FIGURE 9. The structure of the memory cell [76], [77].

TABLE 18. At time model (37 Variables).

Models	Hyperparameter	Accuracy	Precision	Recall	F1-score	Confusion matrix			
Heteropobit		0.785	1: 0.00	1: 0.00	1: 0.00		1	2	3
			2: 0.00	2: 0.00	2: 0.00	1	0	0	51
			3: 0.79	3: 1.00	3: 0.88	2	0	0	14
			WA: 0.62	WA: 0.79	WA: 0.69	3	0	0	238
Kernel SVM	Kernel: RGB Gamma: 0.009 C: 100	0.782	1: 0.33	1: 0.02	1: 0.04		1	2	3
			2: 0.00	2: 0.00	2: 0.00	1	1	1	49
			3: 0.79	3: 0.99	3: 0.88	2	0	0	14
			WA: 0.68	WA: 0.78	WA: 0.70	3	2	0	236
Decision Tree	Criteria: entropy Max depth: 1	0.785	1: 0.00	1: 0.00	1: 0.00		1	2	3
			2: 0.00	2: 0.00	2: 0.00	1	0	0	51
			3: 0.79	3: 1.00	3: 0.88	2	0	0	14
			WA: 0.62	WA: 0.79	WA: 0.69	3	0	0	238
Random Forest	Tree numbers: 3 Depth: 4	0.785	1: 0.00	1: 0.00	1: 0.00		1	2	3
			2: 0.00	2: 0.00	2: 0.00	1	0	0	51
			3: 0.79	3: 1.00	3: 0.88	2	0	0	14
			WA: 0.62	WA: 0.79	WA: 0.69	3	0	0	238
Xgboost	Tree number: 50 Depth: 1 Learning rate: 0.1	0.785	1: 0.00	1: 0.00	1: 0.00		1	2	3
			2: 0.00	2: 0.00	2: 0.00	1	0	0	51
			3: 0.79	3: 1.00	3: 0.88	2	0	0	14
			WA: 0.62	WA: 0.79	WA: 0.69	3	0	0	238
ANN	Epoch: 20 Drop out: .4 Learning rate: 0.00002 Hidden layer 1 node: 148 Hidden layer 2 node: 148	0.785	1: 0.00	1: 0.00	1: 0.00		1	2	3
			2: 0.00	2: 0.00	2: 0.00	1	0	0	51
			3: 0.79	3: 1.00	3: 0.88	2	0	0	14
			WA: 0.62	WA: 0.79	WA: 0.69	3	0	0	238
LSTM	Epoch: 173 Learning rate: 0.0002 Hidden layer node: 74	0.785	1: 0.00	1: 0.00	1: 0.00		1	2	3
			2: 0.00	2: 0.00	2: 0.00	1	0	0	51
			3: 0.79	3: 1.00	3: 0.88	2	0	0	14
			WA: 0.62	WA: 0.79	WA: 0.69	3	0	0	238

B. Word2Vec (W2V)

The Word2Vec (W2V) model is a word distribution-based embedding method and generates dense embedding vectors representing each word's semantic meaning. For example, the W2V model may generate similar embedding vectors for "pen" and "pencil" because the two words contain similar semantic meanings.

As a pre-process, the following steps are applied:

Step 1. Converting emoticon and $ symbols into related words;

TABLE 19. **Ex ante model (59 Variables).**

Models	Hyperparameter	Accuracy	Precision	Recall	F1-score	Confusion matrix			
Heteropobit		0.789	1: 0.57 2: 0.00 3: 0.79 WA: 0.72	1: 0.08 2: 0.00 3: 0.99 WA: 0.79	1: 0.14 2: 0.00 3: 0.88 WA: 0.71		1	2	3
						1	4	0	47
						2	0	0	14
						3	3	0	235
Kernel SVM	Kernel: RGB Gamma: 1.0 C: 0.1	0.785	1: 0.00 2: 0.00 3: 0.79 WA: 0.62	1: 0.00 2: 0.00 3: 10.000. WA: 0.79	1: 0.00 2: 0.00 3: 0.88 WA: 0.69		1	2	3
						1	0	0	51
						2	0	0	14
						3	0	0	238
Decision Tree	Criteria: entropy Max depth: 1	0.785	1: 0.00 2: 0.00 3: 0.79 WA: 0.62	1: 0.00 2: 0.00 3: 10.000. WA: 0.79	1: 0.00 2: 0.00 3: 0.88 WA: 0.69		1	2	3
						1	0	0	51
						2	0	0	14
						3	0	0	238
Random Forest	Tree numbers: 27 Depth: 40	0.785	1: 0.50 2: 0.00 3: 0.80 WA: 0.71	1: 0.12 2: 0.00 3: 0.97 WA: 0.79	1: 0.19 2: 0.00 3: 0.88 **WA: 0.72**		1	2	3
						1	6	0	45
						2	0	0	14
						3	6	0	232
Xgboost	Tree number: 80 Depth: 5 Learning rate: 0.1	0.789	1: 0.60 2: 0.00 3: 0.79 WA: 0.72	1: 0.06 2: 0.00 3: 0.99 WA: 0.79	1: 0.11 2: 0.00 3: 0.88 WA: 0.71		1	2	3
						1	3	0	48
						2	0	0	14
						3	2	0	236
ANN	Epoch: 339 Drop out: 0.4 Learning rate:0.00002 Hidden layer 1 node:148 Hidden layer 2 node: 148	0.776	1: 0.40 2: 0.00 3: 0.79 WA: 0.69	1: 0.08 2: 0.00 3: 0.97 WA: 0.78	1: 0.13 2: 0.00 3: 0.87 WA: 0.71		1	2	3
						1	4	0	47
						2	0	0	14
						3	6	1	231
LSTM	Epoch:399 Learning rate: 0.00002 Hidden layer node:118	0.785	1: 0.00 2: 0.00 3: 0.79 WA: 0.62	1: 0.00 2: 0.00 3: 10.000. WA: 0.79	1: 0.00 2: 0.00 3: 0.88 WA: 0.69		1	2	3
						1	0	0	51
						2	0	0	14
						3	0	0	238

* Notes: WA indicates weighted average macro values. The horizontal labels from 1 (left) to 3 (right) are the predictive classes, while the vertical labels from 1 (top) to 3 (bottom) are the true classes. The values on the diagonal are the number of correct predictions for the star ratings mapped to the horizontal or vertical star ratings.

Step 2. Splitting the review text into words (tokenization);

Step 3. Removing stopwords, punctuation, numbers, and single characters;

Step 4. Lemmatizing words (converting words into the base form, e.g., writing → write).

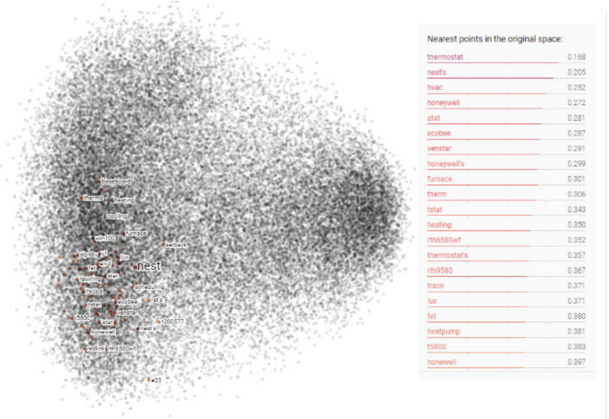
After the above steps, the W2V model generates embedding vectors from each review text. The skip-gram W2V model [52] generates k-dimensional real-vector word embedding v_n for the nth word in all reviews by maximizing the following objective function:

$$\frac{1}{N}\sum_{n=1}^{N}\sum_{-c<s<c;s>0}\log p(word_{n+s}|word_n)$$

where

$$p(word_s|word_n) = \frac{\exp(v'_s v_n)}{\sum_{t=1}^{T}\exp(v'_t v_n)}.$$

where N is the number of words in all the reviews (the entire corpus); c is the window size for selecting neighboring words around the center word n; and T is the number of unique words (vocabulary) in all the reviews. In this study, the W2V model is trained with all the reviews (N = 1,926,047) in the "tool and home improvement" category and the number of unique words is 73,856. The hyperparameters are the W2V embedding dimension, window size, and training dataset. After hyperparameter tuning, the optimal W2V



Nearest points in the original space:

thermostat	0.168
nest's	0.205
hvac	0.252
honeywell	0.272
stat	0.281
ecobee	0.287
venstar	0.291
honeywell's	0.299
furnace	0.301
therm	0.306
tstat	0.343
heating	0.350
rth6580wf	0.352
thermostat's	0.357
rth9580	0.367
trane	0.371
lux	0.371
fet	0.380
heatpump	0.381
t5800	0.383
honewell	0.397

* Notes: https://projector.tensorflow.org/, Initial word2vec dimension is 100 and is reduced to 2 dimensions by applying PCA. X-axis and Y-axis indicate principal component. Distance between words are based on cosine similarity score (=similarity = A • B||A||||B|| where A and B is a vector).

FIGURE 10. **Word2Vector visualization in 2 dimension.**

TABLE 20. Ex ante-sub-model (90 Variables).

Models	Hyperparameter	Accuracy	Precision	Recall	F1-score	Confusion matrix			
Heteropobit		0.789	1: 0.56	1: 0.10	1: 0.17		1	2	3
			2: 0.00	2: 0.00	2: 0.00	1	5	0	46
			3: 0.80	3: 0.98	3: 0.88	2	0	0	14
			WA: 0.72	WA: 0.79	WA: 0.72	3	4	0	234
Kernel SVM	Kernel: RGB Gamma: 1.0 C: 0.1	0.785	1: 0.00	1: 0.00	1: 0.00		1	2	3
			2: 0.00	2: 0.00	2: 0.00	1	0	0	51
			3: 0.79	3: 10.00	3: 0.88	2	0	0	14
			WA: 0.62	WA: 0.79	WA: 0.69	3	0	0	238
Decision Tree	Class weighted Criteria: entropy Max depth: 1	0.785	1: 0.00	1: 0.00	1: 0.00		1	2	3
			2: 0.00	2: 0.00	2: 0.00	1	0	0	51
			3: 0.79	3: 10.00	3: 0.88	2	0	0	14
			WA: 0.62	WA: 0.79	WA: 0.69	3	0	0	238
Random Forest	Tree numbers: 16 Depth: 42	**0.802**	1: 0.73	1: 0.16	1: 0.26		1	2	3
			2: 0.00	2: 0.00	2: 0.00	1	6	2	43
			3: 0.80	3: 0.99	3: 0.89	2	0	0	14
			WA: 0.75	WA: 0.80	**WA: 0.74**	3	5	0	233
Xgboost	Tree number: 100 Depth: 4 Learning rate: 0.2	**0.802**	1: 0.78	1: 0.14	1: 0.23		1	2	3
			2: 0.00	2: 0.00	2: 0.00	1	7	0	44
			3: 0.80	3: 0.99	3: 0.89	2	0	0	14
			WA: 0.76	WA: 0.80	**WA: 0.74**		2	0	236
ANN	Epoch: 3 Drop out: .4 Learning rate: 0.0002 Hidden layer 1 node:180 Hidden layer 2 node:180	0.782	1: 0.38	1: 0.06	1: 0.10		1	2	3
			2: 0.00	2: 0.00	2: 0.00	1	3	0	48
			3: 0.79	3: 0.98	3: 0.88	2	1	0	13
			WA: 0.69	WA: 0.78	WA: 0.71	3	4	0	234
LSTM	Epoch: 232 Learning rate: 0.0002 Hidden layer node: 322	0.700	1: 0.50	1: 0.02	1: 0.04		1	2	3
			2: 0.00	2: 0.00	2: 0.00	1	1	0	50
			3: 0.79	3: 10.00	3: 0.88	2	0	0	14
			WA: 0.70	WA: 0.79	WA: 0.70	3	1	0	237

TABLE 21. Model 4: Ex ante-sub-price model (94 Variables).

Models	Hyperparameter	Accuracy	Precision	Recall	F1-score	Confusion matrix			
Kernel SVM	Kernel: RGB Gamma: 1.0 C: .1	0.785	1: 0.00	1: 0.00	1: 0.00		1	2	3
			2: 0.00	2: 0.00	2: 0.00	1	0	0	51
			3: 0.79	3: 10.00	3: 0.88	2	0	0	14
			WA: 0.62	WA: 0.79	WA: 0.69		0	0	238
Decision Tree	Class weighted Criteria: entropy Max depth: 1	0.785	1: 0.00	1: 0.00	1: 0.00		1	2	3
			2: 0.00	2: 0.00	2: 0.00	1	0	0	51
			3: 0.79	3: 10.00	3: 0.88	2	0	0	14
			WA: 0.62	WA: 0.79	WA: 0.69		0	0	238
Random Forest	Class weighted Tree numbers: 17 Depth: 43	0.785	1: 0.50	1: 0.06	1: 0.11		1	2	3
			2: 0.00	2: 0.00	2: 0.00	1	3	0	48
			3: 0.79	3: 0.99	3: 0.88	2	0	0	14
			WA: 0.71	WA: 0.79	WA: 0.71	3	3	0	235
Xgboost	Tree number: 100 Depth: 1 Learning rate: 0.4	0.782	1: 0.00	1: 0.00	1: 0.00		1	2	3
			2: 0.00	2: 0.00	2: 0.00	1	0	0	51
			3: 0.78	3: 10.00	3: 0.88	2	0	0	14
			WA: 0.62	WA: 0.78	WA: 0.69	3	1	0	237
ANN	Class weighted Epoch: 999 Drop out: 0.4 Learning rate:.0.00005 Hidden layer 1 node:282 Hidden layer 2 node:188	0.792	1: 0.75	1: 0.06	1: 0.11		1	2	3
			2: 0.00	2: 0.00	2: 0.00	1	3	0	48
			3: 0.79	3: 10.00	3: 0.88	2	0	0	14
			WA: 0.75	WA: 0.79	WA: 0.71	3	1	0	237
LSTM	Epoch: 22 Learning rate: 0.0002 Hidden layer node: 376	0.769	1: 0.35	1: 0.12	1: 0.18		1	2	3
			2: 0.00	2: 0.00	2: 0.00	1	6	0	45
			3: 0.79	3: 0.95	3: 0.87	2	0	0	14
			WA: 0.68	WA: 0.77	WA: 0.71	3	11	0	227

* Notes: Heteroskedastic ordered probit (HETOP) is excluded because the model is incompatible with product dummies due to multicollinearity. In addition, Class weight $= \frac{N}{K \times N_k}$, where N is the number of samples; K is number of classes; and, N_k *is* the number of samples belong to class k. Class weight values [0.427, 5.09, 2.16] for each class (3, 2, and 1 class)

embedding dimension is 100 and the optimal window size is 5.

C. BIDIRECTIONAL ENCODER REPRESENTATIONS FROM TRANSFORMERS (BERT)

Bidirectional Encoder Representations from Transformers (BERT) is a state-of-the-art context-based embedding method. BERT can represent the same word in a sentence with different embedding vectors by reflecting the contextual meaning of each word in the sentence. For example, in the sentences "I did not like this thermostat in the past. Now, I love this thermostat," the word "thermostat" occurs twice, in the first and in the second sentence. BERT generates different embedding vectors for "thermostat" in the first and second sentences based on the contextual information in them. Meanwhile, context-free embedding models

TABLE 22. Model 5: Partial ex ante-sub-model (161 Variables).

Models	Hyperparameter	Accuracy	Precision	Recall	F1-score	Confusion matrix			
Kernel SVM	Kernel: RGB Gamma: 1.0 C: 0.1	0.785	1: 0.00	1: 0.00	1: 0.00		1	2	3
			2: 0.00	2: 0.00	2: 0.00	1	0	0	51
			3: 0.79	3: 10.00	3: 0.88	2	0	0	14
			WA: 0.62	WA: 0.79	WA: 0.69	3	0	0	238
Decision Tree	Criteria: Entropy Max depth: 4	0.779	1: 0.25	1: 0.02	1: 0.04		1	2	3
			2: 0.00	2: 0.00	2: 0.00	1	1	0	50
			3: 0.79	3: 0.99	3: 0.88	2	0	0	14
			WA: 0.66	WA: 0.78	WA: 0.69	3	3	0	235
Random Forest	Tree numbers: 11 Depth: 14	0.789	1: 0.57	1: 0.08	1: 0.14		1	2	3
			2: 0.00	2: 0.00	2: 0.00	1	4	0	47
			3: 0.79	3: 0.99	3: 0.88	2	0	0	14
			WA: 0.72	WA: 0.79	WA: 0.71	3	3	0	235
Xgboost	Class weighted Tree number: 90 Depth: 2 Learning rate:.1	0.759	1: 0.42	1: 0.41	1: 0.42		1	2	3
			2: 0.00	2: 0.00	2: 0.00	1	21	0	30
			3: 0.83	3: 0.88	3: 0.85	2	0	0	14
			WA: 0.72	WA: 0.76	**WA: 0.74**	3	29	0	209
ANN	Epoch: 196 Drop out: 0.3 Learning rate: 0.0002 Hidden layer 1 node: 322 Hidden layer 2 node: 322	0.792	1: 0.75	1: 0.06	1: 0.11		1	2	3
			2: 0.00	2: 0.00	2: 0.00	1	3	0	48
			3: 0.79	3: 10.00	3: 0.88	2	0	0	14
			WA: 0.75	WA: 0.79	WA: 0.71	3	1	0	237
LSTM	Epoch: 127 Learning rate: 0.0002 Hidden layer node: 242	0.789	1: 10.00	1: 0.02	1: 0.04		1	2	3
			2: 0.00	2: 0.00	2: 0.00	1	1	0	50
			3: 0.79	3: 10.00	3: 0.88	2	0	0	14
			WA: 0.79	WA: 0.79	WA: 0.70	3	0	0	238

TABLE 23. Model 6: Partial ex ante-sub-price model (165 Variables).

Models	Hyperparameter	Accuracy	Precision	Recall	F1-score	Confusion matrix			
Kernel SVM	Kernel: RGB Gamma: 1.0 C: 0.1	0.785	1: 0.00	1: 0.00	1: 0.00		1	2	3
			2: 0.00	2: 0.00	2: 0.00	1	0	0	51
			3: 0.79	3: 10.00	3: 0.88	2	0	0	14
			WA: 0.62	WA: 0.79	WA: 0.69	3	0	0	238
Decision Tree	Class weighted Criteria: entropy Max depth: 4	0.785	1: 0.00	1: 0.00	1: 0.00		1	2	3
			2: 0.00	2: 0.00	2: 0.00	1	0	0	51
			3: 0.79	3: 10.00	3: 0.88	2	0	0	14
			WA: 0.62	WA: 0.79	WA: 0.69	3	0	0	238
Random Forest	Tree numbers: 23 Depth: 15	0.799	1: 0.70	1: 0.14	1: 0.23		1	2	3
			2: 0.00	2: 0.00	2: 0.00	1	7	0	44
			3: 0.80	3: 0.99	3: 0.89	2	0	0	14
			WA: 0.75	WA: 0.80	WA: 0.73	3	3	0	235
Xgboost	Tree number: 50 Depth:12 Learning rate: 0.1	0.795	1: 0.56	1: 0.20	1: 0.29		1	2	3
			2: 0.00	2: 0.00	2: 0.00	1	10	0	41
			3: 0.81	3: 0.97	3: 0.88	2	1	0	13
			WA: 0.73	WA: 0.80	WA: 0.74	3	7	0	231
ANN	Epoch: 597 Drop out: 0.4 Learning rate: 0.00002 Hidden layer 1 node:495 Hidden layer 2 node: 495	0.792	1: 0.67	1: 0.08	1: 0.14		1	2	3
			2: 0.00	2: 0.00	2: 0.00	1	4	0	47
			3: 0.79	3: 0.99	3: 0.88	2	0	0	14
			WA: 0.74	WA: 0.79	WA: 0.72	3	2	0	236
LSTM	Epoch:1945 Learning rate: 0.00002 Hidden layer node:660	0.782	1: 0.33	1: 0.02	1: 0.04		1	2	3
			2: 0.00	2: 0.00	2: 0.00	1	1	0	50
			3: 0.79	3: 0.99	3: 0.88	2	0	0	14
			WA: 0.67	WA: 0.78	WA: 0.70	3	2	0	236

(e.g., TF-IDF and W2V) generate the same embedding vectors for "thermostat" in both sentences.

In particular, the domain expert in this study reads and annotates all 5,307 reviews for PTs and finds that the review text often contains a comparison between the previously owned PT and the newly purchased PT; therefore, the same word in the review often represents different contexts based on its position in the review. For example, "I disliked the previous thermostat. However, I love this new thermostat." In this text, even though the word "thermostat" occurs both in the first and in the second sentence, the first one may contain a negative sentiment and the second one may contain a positive sentiment.

However, context-free embedding models (e.g., TF-IDF and W2V) cannot capture different semantic meanings of the same word in different positions in the review sentences. In contrast to the context-free embedding models, BERT (context-based embedding) can find the contextual difference between occurrences of the same word in different positions in the review sentences.

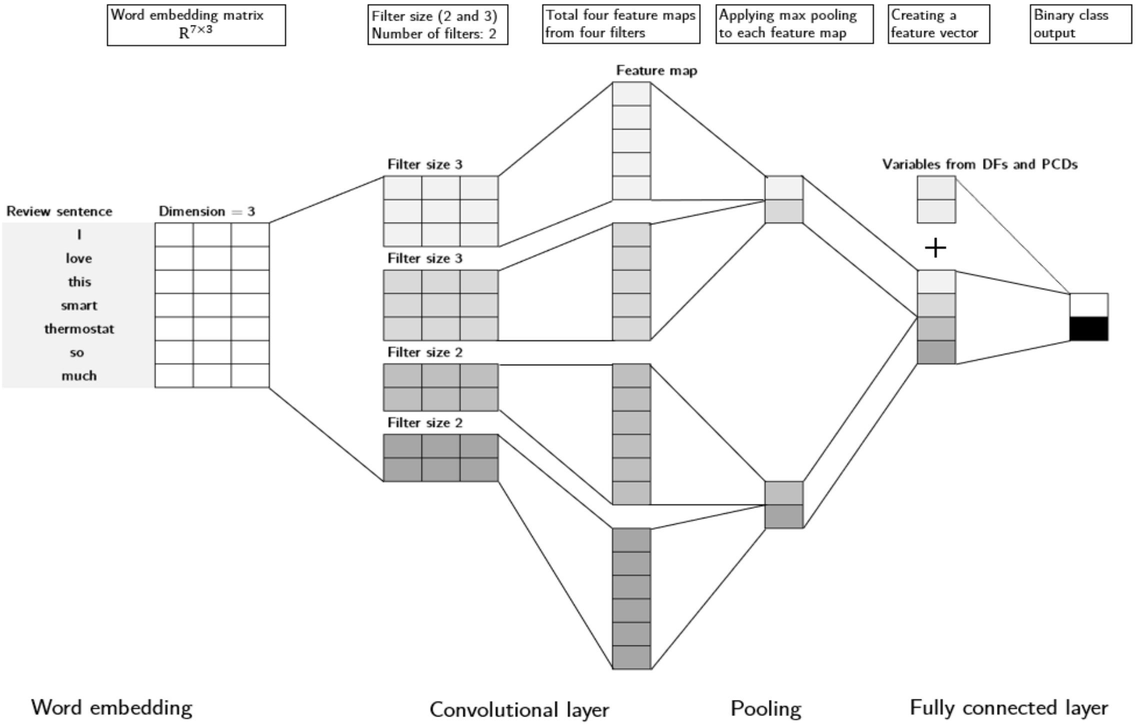

Word embedding matrix $R^{7\times3}$
Filter size (2 and 3) Number of filters: 2
Total four feature maps from four filters
Applying max pooling to each feature map
Creating a feature vector
Binary class output
Feature map
Filter size 3
Variables from DFs and PCDs
Review sentence
Dimension = 3
I
love
this
smart
thermostat
so
much
Filter size 3
+
Filter size 2
Filter size 2
Word embedding
Convolutional layer
Pooling
Fully connected layer

FIGURE 11. The structure of the CNN [55], [56].

The pre-trained BERT embedding model is trained with 800 million words using a book corpus [79] and 2,500 million words from Wikipedia data. BERT uses the WordPiece tokenizer [80], which splits each word into sub-words to deal with out-of-vocabulary words.

BERT's structure is based on multilayered transformer encoders [81]. BERT is trained for two objectives: masked language modeling (MLM) and next sentence prediction (NSP). MLM is a prediction task for randomly masked tokens in the sentences to learn about the contextual information in the text. NSP is a binary classification indicating whether the second sentence is a subsequent sentence to the first one to learn about the relationship between sentences.

This study uses the BERT-based model, which contains 30,522 unique tokens with 768 embedding dimensions for fine-tuning and further pre-training. With a fine-tuned BERT, the CNN is applied on top of the pre-trained embedding from the original BERT model. Having further pre-trained BERT, the BERT embedding is updated by training on the review text data and is used as input vectors for the CNN classifier. Recently, Gururangan *et al.* [51] and Sun *et al.* [53] showed that further pre-training with domain data could improve machine learning models' performance.

APPENDIX I
CONVOLUTIONAL NEURAL NETWORK (CNN) FOR SENTIMENT CLASSIFICATION

Figure 11 provides an example of a simplified CNN model for the binary classification model. The structure of the CNN in this example has four layers. The first layer is the input word embedding generated from the review text. Each review text is split into tokens (e.g., words in a W2V model and sub-words in a BERT model) and becomes a sequence of the tokens with length n. The tokenized review is denoted as $x_{1:n}$. Each token x_i is mapped to a word-embedding vector R^d. The embedded sequence of tokens $x_{1:n}$ is expressed as follows:

$$x_{1:n} = x_1 \oplus x_2 .. \oplus x_n, \quad \text{where } x_i \in R^d, \ i \in \{1, \ldots, n\}$$

and where each class has a predicted probability, and the class showing the highest predicted probability will be the predicted class.

ACKNOWLEDGMENT

This article is based on the first chapter of his Ph.D. dissertation. The preliminary results were presented in the poster sections at the 2021 American Economic Association, the 2020 National Association for Business Economics Tech conference, and the 2020 Rocky Mountain Advanced Computing Consortium. This article was posted in the 2020 Job Market Papers section on Research Papers in Economics (RePEc). Proofreading of this article was completed with Proof-Reading-Service.com. In addition, the author thank Alan Love, Jinhui Bai, Bidisha Mandal, all professors, staff, and students at WSU SES, Melanie Lynette Thongs, Christopher Steven Duke, Roy Obenchain, Peter Mills at WSU. He also thank Julian McAuley (UCSD) to share the dataset in this study. All errors in this article are related to him alone.

REFERENCES

[1] A. Timoshenko and J. R. Hauser, “Identifying customer needs from user-generated content,” *Marketing Sci.*, vol. 38, no. 1, pp. 1–20, Jan. 2019.

[2] X. Liu, D. Lee, and K. Srinivasan, “Large-scale cross-category analysis of consumer review content on sales conversion leveraging deep learning,” *J. Marketing Res.*, vol. 46, no. 6, pp. 918–943, 2019.

[3] S. J. Mäkinen, “Internet-of-Things disrupting business ecosystems: A case in home automation,” in *Proc. IEEE Int. Conf. Ind. Eng. Eng. Manag.*, Dec. 2014, pp. 1467–1470.

[4] R. Yang and M. W. Newman, “Living with an intelligent thermostat: Advanced control for heating and cooling systems,” in *Proc. ACM Conf. Ubiquitous Comput. (UbiComp)*, 2012, pp. 1102–1107.

[5] K. E. Train, *Discrete Choice Methods With Simulation*. Cambridge, U.K.: Cambridge Univ. Press, 2009.

[6] S. Tadelis, “Reputation and feedback systems in online platform markets,” *Annu. Rev. Econ.*, vol. 8, pp. 321–340, Oct. 2016.

[7] G. Cui, H.-K. Lui, and X. Guo, “The effect of online consumer reviews on new product sales,” *Int. J. Electron. Commerce*, vol. 17, no. 1, pp. 39–58, Oct. 2012.

[8] R. He and J. McAuley, “Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering,” in *Proc. 25th Int. Conf. World Wide Web*, Apr. 2016, pp. 507–517.

[9] M. Anderson and J. Magruder, “Learning from the crowd: Regression discontinuity estimates of the effects of an online review database,” *Econ. J.*, vol. 122, no. 563, pp. 957–989, 2012.

[10] Y. Chen. (2018). *User-Generated Physician Ratings: Evidence From Yelp*. [Online]. Available: https://www. softwareadvice.com/ resources/how-patientsuse

[11] J. A. Chevalier and D. Mayzlin, “The effect of word of mouth on sales: Online book reviews,” *J. Mark. Res.*, vol. 43, no. 3, pp. 345–354, 2006.

[12] N. Hu, L. Liu, and J. J. Zhang, “Do online reviews affect product sales? The role of reviewer characteristics and temporal effects,” *Inf. Technol. Manage.*, vol. 9, no. 3, pp. 201–214, Sep. 2008.

[13] M. Luca, “Reviews, reputation, and revenue: The case of Yelp.com,” *Harvard Bus. School NOM Unit Working Paper*, to be published.

[14] D. Mayzlin, Y. Dover, and J. Chevalier, “Promotional reviews: An empirical investigation of online review manipulation,” *Amer. Econ. Rev.*, vol. 104, no. 8, pp. 2421–2444, 2014.

[15] I. C. Reimers and J. Waldfogel, “Digitization and pre-purchase information: The causal and welfare impacts of reviews and crowd ratings,” *Amer. Econ. Rev.*, vol. 111, no. 6, pp. 1944–1971, 2020.

[16] M. Luca, “Designing online marketplaces: Trust and reputation mechanisms,” *Innov. Policy Economy*, vol. 17, pp. 77–93, Jan. 2017.

[17] M. Luca and G. Zervas, “Fake it till you make it: Reputation, competition, and yelp review fraud,” *Manag. Sci.*, vol. 62, no. 12, pp. 3412–3427, 2016.

[18] Y. Zhao, S. Yang, V. Narayan, and Y. Zhao, “Modeling consumer learning from online product reviews,” *Mark. Sci*, vol. 32, no. 1, pp. 153–169, 2013.

[19] G. Donaker, H. Kim, M. Luca, and M. Weber, “Designing better online review systems,” *Harvard Bus. Rev.*, vol. 97, no. 6, pp. 122–129, 2019.

[20] M. M. Susan and S. David, “What makes a helpful online review? A study of customer reviews on amazon.com,” *MIS Quart.*, vol. 34, no. 1, pp. 185–200, 2010.

[21] N. Hu, P. A. Pavlou, and J. Zhang, “Can online reviews reveal a product’s true quality: Empirical findings and analytical modeling of online word-of-mouth communication,” in *Proc. 7th ACM Conf. Electron. Commerce (EC)*, 2006, pp. 324–330.

[22] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent Dirichlet allocation,” *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.

[23] R. J. Passonneau, C. Rudin, A. Radeva, and Z. A. Liu, “Reducing noise in labels and features for a real world dataset: Application of nlp corpus annotation methods,” in *Proc. Int. Conf. Intell. Text Process. Comput. Linguist.*, 2009, pp. 86–97.

[24] W. H. Greene, *Econometric Analysis*. 7th ed. Harlow, U.K.: Pearson Education, 2012.

[25] S. Chen and S. Khan, “Rates of convergence for estimating regression coefficients in heteroskedastic discrete response models,” *J. Econom.*, vol. 117, no. 2, pp. 245–278, 2003.

[26] R. Williams, “Using heterogeneous choice models to compare logit and probit coefficients across groups,” *Sociol. Methods Res.*, vol. 37, no. 4, pp. 531–559, 2009.

[27] W. H. Greene and D. A. Hensher, *Modeling Ordered Choices: A Primer*. Cambridge, U.K.: Cambridge Univ. Press, 2010.

[28] T. D. Chen and K. M. Kockelman, “Roles of vehicle footprint, height, and weight in crash outcomes: Application of a heteroscedastic ordered probit model,” *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2280, no. 1, pp. 89–99, Jan. 2012.

[29] W. H. Greene and D. A. Hensher, “Ordered choices and heterogeneity in attribute processing,” *J. Transp. Econ. Policy*, vol. 44, no. 3, pp. 331–364, 2010.

[30] L. Keele and D. K. Park, “Difficult choices: An evaluation of heterogeneous choice models,” in *Proc. Paper Meeting Amer. Political Sci. Assoc.*, 2006, pp. 2–5.

[31] J. D. Lemp, K. M. Kockelman, and A. Unnikrishnan, “Analysis of large truck crash severity using heteroskedastic ordered probit models,” *Accid. Anal. Prev.*, vol. 43, no. 1, pp. 370—380, 2011.

[32] J. Litchfield, B. Reilly, and M. Veneziani, “An analysis of life satisfaction in albania: An heteroscedastic ordered probit model approach,” *J. Econ. Behav. Organ.*, vol. 81, no. 3, pp. 731–741, 2012.

[33] X. Wang and K. M. Kockelman, “Use of heteroscedastic ordered logit model to study severity of occupant injury: Distinguishing effects of vehicle weight and type,” *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1908, no. 1, pp. 195–204, Jan. 2005.

[34] M. Bouazizi and T. Ohtsuki, “Multi-class sentiment analysis on twitter: Classification performance and challenges,” *Big Data Mining Anal.*, vol. 2, no. 3, pp. 181–194, 2019.

[35] T. U. Haque, N. N. Saber, and F. M. Shah, “Sentiment analysis on large scale Amazon product reviews,” in *Proc. IEEE Int. Conf. Innov. Res. Develop. (ICIRD)*, May 2018, pp. 1–6.

[36] C. Chen, A. Liaw, and L. Breiman, “Using random forest to learn imbalanced data,” Univ. California, Berkeley, CA, USA, Tech. Rep. 666, 2004.

[37] T. Dietterich, “Overfitting and undercomputing in machine learning,” *ACM Comput. Surv.*, vol. 27, no. 3, pp. 326–327, 1995.

[38] P. B. Schiilkop, C. Burgest, and V. Vapnik, “Extracting support data for a given task,” in *Proc. Int. Conf. Data Min. Knowl. Discov.*, Menlo Park, CA, USA: AAAI Press, 1995, pp. 252–257.

[39] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning*, vol. 112. New York, NY, USA: Springer, 2013.

[40] T. G. Dietterich, “Ensemble methods in machine learning,” in *Proc. Int. Workshop Multiple Classifier Syst.* Berlin, Germany: Springer, 2000, pp. 1–15.

[41] J. Friedman, T. Hastie, and R. Tibshirani, *The Elements of Statistical Learning*, vol. 1, no. 10. New York, NY, USA: Springer, 2001.

[42] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 785–794.

[43] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, pp. 436–444, May 2015.

[44] S. M. Ramsey and J. S. Bergtold, “Examining inferences from neural network estimators of binary choice processes: Marginal effects, and willingness-to-pay,” *Comput. Econ.*, pp. 1–29, Jun. 2020.

[45] S. Hochreiter and J. J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[46] M. Ibrahim, M. Torki, and N. El-Makky, “Imbalanced toxic comments classification using data augmentation and deep learning,” in *Proc. 17th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2018, pp. 875–878.

[47] L. A. Jeni, J. F. Cohn, and F. D. L. Torre, “Facing imbalanced data recommendations for the use of performance metrics,” in *Proc. Humaine Assoc. Conf. Affect. Comput. Intel. Interact.*, 2013, pp. 245–251.

[48] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” 2018, *arXiv:1810.04805*.

[49] D. Erhan, A. Courville, Y. Bengio, and P. Vincent, “Why does unsupervised pre-training help deep learning,” in *Proc. Int. Conf. Artif. Intell. Stat.*, 2010, pp. 201–208.

[50] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2009.

[51] S. Gururangan, A. Marasović, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, and N. A. Smith, “Don’t stop pretraining: Adapt language models to domains and tasks,” 2020, *arXiv:2004.10964*.

[52] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Proc. Adv. Neural Inf. Process. Syst (NIPS)*, 201, pp. 31113–31119.

[53] C. Sun, X. Qiu, Y. Xu, and X. Huang, “How to fine-tune BERT for text classification,” in *Proc. China Nat. Conf. Chin. Comput. Linguist.*, 2019, pp. 194–206.

[54] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics (Long Papers)*, vol. 1, 2014, pp. 655–665.

[55] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1746–1751.

[56] Y. Zhang and B. Wallace, "A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification," 2015, *arXiv:1510.03820*.

[57] D. A. Garvin, "What does product quality really mean," *Sloan Manag. Rev.*, vol. 25, pp. 25–43, Oct. 1984.

[58] S. Syed and M. Spruit, "Full-text or abstract? Examining topic coherence scores using latent Dirichlet allocation," in *Proc. IEEE Int. Conf. Data Sci. Adv. Anal. (DSAA)*, Oct. 2017, pp. 74–164.

[59] D. Mallick, "Marginal and interaction effects in ordered response models," *MPRA Paper*, to be published.

[60] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[61] T. Papadimitriou, P. Gogas, and E. Stathakis, "Forecasting energy markets using support vector machines," *Energy Econ.*, vol. 44, pp. 135–142, Jul. 2014.

[62] J.-P. Vert, K. Tsuda, and B. Schölkopf, "A primer on kernel methods," *Kernel Methods Comput. Biol.*, vol. 47, pp. 35–70, Dec. 2004.

[63] M. Pal, "Multiclass approaches for support vector machine based land cover classification," 2008, *arXiv:0802.2411*.

[64] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[65] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, Oct. 2001.

[66] W. Chen, K. Fu, J. Zuo, X. Zheng, T. Huang, and W. Ren, "Radar emitter classification for large data set based on weighted-xgboost," *IET Radar, Sonar Navigat.*, vol. 11, no. 8, pp. 1203–1207, 2017.

[67] G. L. Urban, A. Timoshenko, P. S. Dhillon, and J. R. Hauser, "Is deep learning a game changer for marketing analytics," *MIT Sloan Manag. Rev.*, vol. 61, no. 2, pp. 70–76, 2020.

[68] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Math. Control Signals Syst.*, vol. 2, no. 4, pp. 303–314, 1989.

[69] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Netw.*, vol. 4, no. 2, pp. 251–257, 1991.

[70] G. Strang, *Linear Algebra and Learning From Data*. Wellesley, MA, USA: Wellesley-Cambridge Press, 2019.

[71] C. M. Kuan and H. White, "Artificial neural networks: An econometric perspective," *Econom. Rev.*, vol. 13, no. 1, pp. 1–91, 1994.

[72] Y. Chauvin and D. E. Rumelhart, *Backpropagation: Theory, Architectures, and Applications*. Hove, U.K.: Psychology Press, 1995.

[73] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[74] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *Int. J. Uncertain. Fuzziness Knowlege-Based Syst.*, vol. 6, no. 2, pp. 107–116, 1998.

[75] G. Rao, W. Huang, Z. Feng, and Q. Cong, "LSTM with sentence representations for document-level sentiment classification," *Neurocomputing*, vol. 308, pp. 49–57, Sep. 2018.

[76] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *Eur. J. Oper. Res.*, vol. 270, no. 2, pp. 654–659, 2018.

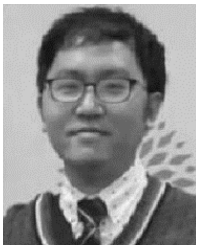
[77] C. Olah. (2015). *Understanding LSTM Networks*. [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[78] M. Gentzkow, B. Kelly, and M. Taddy, "Text as data," *J. Econ. Lit.*, vol. 57, no. 3, pp. 535–574, 2019.

[79] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler, "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 19–27.

[80] Y. Wu *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," 2016, *arXiv:1609.08144*.

[81] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.



JIKHAN JEONG received the B.Sc. degree in chemical engineering from Sungkyunkwan University (SKKU), Suwon, South Korea, in 2010, the M.Sc. degree in management science from KAIST, Daejeon, South Korea, in 2013, and the Ph.D. degree in economics from Washington State University, WA, USA, in 2021. He was an Associate Research Fellow with the Management Research Institute, Korea Electric Power Corporation (KEPCO), from December 2012 to October 2021. He is currently a Visiting Assistant Professor with the Department of Data Science and Business Analytics, Florida Polytechnic University (FPU). His research interests include digital economics, marketing analytics using AI and big data, fintech, machine learning, natural language processing, and applied econometrics.