IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 2 November 2022, accepted 26 November 2022, date of publication 30 November 2022,
date of current version 5 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3225558

RESEARCH ARTICLE

Distribution-Adapted Model for Helpful Vote Prediction

RISTU SAPTONO[1,2], (Member, IEEE), AND TSUNENORI MINE[1]
[1]Department of Advanced Information Technology, Kyushu University, Fukuoka 819-0395, Japan
[2]Department of Informatics, Universitas Sebelas Maret, Surakarta 57126, Indonesia

Corresponding author: Ristu Saptono (ristu.saptono@staff.uns.ac.id)

This work was supported in part by the Japan Society for the Promotion of Science (JSPS) KAKENHI under Grant JP21H00907, Grant JP21K11847, Grant JP20H01728, Grant JP20H04300, and Grant JP19KK0257; and in part by the Universitas Sebelas Maret PDD under Grant 260/UN27.22/HK.07.00/2021 and Grant 254/UN27.22/PT.01.03/2022.

ABSTRACT The number of helpful votes on a review is an essential indicator of how much impact the review has on other customers in electronic commerce. Therefore, predicting the number of helpful votes is an important task. Regression analysis and Tobit modeling are typical methods of prediction. Those methods come from the same initial assumption that the number of helpful votes follows a normal distribution on any dataset. However, the assumption is not usually confirmed, and the distribution of the helpful votes often follows other distributions. This paper proposes a framework for investigating the feasibility of building a model that predicts the number of helpful votes according to the distribution of the number of helpful votes. On top of that, considering the review age, we propose an adaptive window size sampling method to evaluate the model on review datasets sorted chronologically. The experimental results validated that the model adapting to the best approximate distribution gives a significant improvement compared to the baseline models. In addition, model evaluation using the adaptive window size sampling method has significant impacts on the performance on large datasets.

INDEX TERMS Distribution-adapted model, adaptive windows size sampling method, helpful vote.

I. INTRODUCTION

The customer often writes a review to describe their opinion about the quality of a product. This review might help other customers with their purchase decision. The number of helpful votes in a product review indicates the impact of a review has on other customers. Hence, it is crucial to estimate the number of helpful votes.

Previous studies examine the distribution of helpful votes[1] in selecting a suitable model by some simple indicators. Negative binomial regression [1], [2], [3] is chosen instead of Poisson regression because they consider that helpful votes are in a count distribution with an over-dispersion problem [1]. Over-dispersion is a phenomenon where the equality of mean and variance is not fulfilled in a count distribution. For the same reason, some studies employ the regression [4], [5] and Tobit model [6], [7] by first taking a normalization or transformation of helpful votes. Normalization and transformation, such as helpful ratio [1], [8], [9] and log-transformation [5], are used to take the helpful votes into a continuous distribution form. However, it has never been confirmed that the distribution initially assumed by the model conforms to or even approximate the distribution of helpful votes. If regression models above are applied to an unsuitable distribution, it may not achieve optimal results and even not be acceptable.

The importance of confirming the target distribution has been introduced to optimize the result on a normal distribution with Gaussian process [10]. The generalized linear model (GLM) provides a solution when the target is not in a normal distribution. The main idea of GLM is to build a model by generalizing regression analysis to other distributions that fit the target [11], [12]. The next problem is to provide the target distribution before developing a GLM.

The goodness of fit test is usually performed to find the best fit data distribution. However, the goodness of fit

The associate editor coordinating the review of this manuscript and approving it for publication was Taous Meriem Laleg-Kirati.

[1]We use this expression 'the distribution of helpful votes' in the same meaning as 'the distribution of the number of helpful votes'.

test cannot often find the best fit distribution. Therefore, we use Mean Squared Error (MSE) and Akaike Information Criterion (AIC) [13], [14] to approximate the distribution by comparing the score of several distributions. Since the normal distribution is in the Exponential Dispersion Model (EDM) family, we use four other distributions: Gamma, Inverse Gaussian (InvGauss), Exponential (Expon), and Wald.

The critical step to approximate the distribution by AIC and MSE is to create a histogram with a certain number of bins. Any constant is often applied as the number of bins without considering dataset characteristics, which lead to wrong identification of the distributions. In this study, we apply Scott's rule [15] to calculate the number of bins and use the Kolmogorov-Smirnov (KS) score [16], [17], [18] for calibration. Later, we also investigate the possibility of the model performance following the rank of distributions identified.

Subsequently, we generate and evaluate the model on the dataset by using a sampling method. Cross-validation with 10-fold sampling is popularly employed to evaluate the helpful-review models [3], [9]. However, a new review under actual conditions does not have any votes yet when posted. Saptono and Mine [19] proposed time-based sampling (TBS) methods with Cochran's formula, which assumes a binomial distribution for classification tasks. Their formula uses the binomial variance of the helpfulness rating calculated from the whole dataset. Besides, only data in the training set are assumed labeled, and the others are unlabeled. Here, the variance formula for the binomial distribution is changed to that for the other distributions so that the TBS method can be used correctly and more effectively.

To address the problems described above, we propose a framework to correctly implement a model adapting to the distribution of helpful votes. Our framework collaborates three main modules: distribution identification, model generation, and sampling methods. Each module employs a particular technique and contributes as follows:

1) We propose a method for identifying the distribution of the helpful votes. The proposed method approximates the distribution in more detail by computing MSE and AIC scores by means of a histogram whose bin counts are computed by Scott's rule. We apply the KS score for calibration.
2) On the model generation, we employ a model adapting to the distribution of helpful votes to predict the number of helpful votes. We call the model the distribution-adapted model. We build the model in three machine learning models: linear model, extreme gradient boosting [20], and convolutional neural network [9], [21].
3) On the sampling methods to evaluate the models, we adjust the window size of the TBS method [19] so that it can be applied to a dataset even in a continuous distribution.

Next, we conduct extensive experiments on Amazon.com datasets [22] and IMDb datasets [23].

In this paper, we answer the following research questions:

Q1 Does distributional identification by MSE or AIC score yield the same results as the KS score?
Q2 Does the performance of the distribution-adapted model follow the rank of distribution identification results?
Q3 Does the adjustment of window size of the TBS method improve the model performance?
Q4 How are the effect of the implementation and evaluation to the time consumption of the distribution-adapted model with the AWS sampling method compared to baseline models?

The rest of the paper is structured as follows: we present an overview of existing prediction models, factors, and sampling methods to estimate the helpful votes in Section II and the typical structure of the EDM family density function in Section III. Subsequently, we elaborate on our proposed framework in Section IV. In Section V, we describe our experimental setup and report the results. Finally, we summarize our contributions and discuss further tasks in Section VI.

II. RELATED WORK

In this section, we briefly describe some previous studies related to ours. We first discuss some metrics to measure helpfulness and then describe some models employed in helpful vote prediction. Subsequently, we discuss some factors used in previous research projects and trends to use the text factor. We next elaborate on previously implemented sampling methods. Finally, we summarize related studies and compare them with this study, as shown in Table 1.

A. HELPFULNESS METRICS

The previous paper used some metrics to measure how helpful the review is for the customer. A helpfulness rating also called a helpful ratio, is applied if there are two types of feedback captured by the system: helpful and not helpful [8], [25], [26], [35], [36]. In this case, the helpfulness rating is a ratio of the number of helpful votes to the total votes. A higher helpfulness rating means the review has helped other customers to make a purchasing decision. Amazon.com also used this metric on their dataset [37]. This metric is also used to binarize the helpfulness rating with a threshold [4].

Recently, most commerce systems, including Amazon.com and Yelp.com, have eliminated the unhelpful button as customer feedback for product review. Consequently, the current Amazon.com 2018 dataset [22], the updated version of the previous Amazon.com 2014 dataset [37], has dropped the total votes information and provides the number of helpful votes as the only feature indicating helpfulness. Previous research used the helpful votes to represent the number of helpful votes [4], [6], [7]. Moreover, the categorized helpful vote form is also used as a target variable [3], [19], [33], [34].

B. HELPFUL VOTE PREDICTION MODELS

Regression analysis is a representative model for predicting review helpfulness [4], [9], [24], [26], especially in terms of

TABLE 1. Related research on helpful vote identification.

Dependent Variable	Independent Variable	Data Source	Sampling Method	Model
R1 helpfulness rating [1], [7]–[9], [24]–[30] R2 binary categorized helpfulness rating [4]	F1 star rating [1], [4], [7], [8], [25]–[29], F2 review length/words [1], [4], [7], [8], [25]–[29], F3 review length/sentences [26], F4 review text [9], [24], F5 review age/days [1], [29], [30], F6 total votes [4], [8], [25], [29], F7 review order [4], [27], F8 text sentiment [1], [27], [29], F9 sentiment score deviation [4] F10 text complexity [1], F11 readability/The automated readability index [4], [27], [29], F12 review types [1], F13 writing style [27], F14 review uncertainty [29], F15 information-entropy increment [4], F16 emotion (anger, sad, sadness) [25], F17 the average number of words of each sentence [26], F18 ratio of type of words (adverb, noun, preposition) [26] and adjective [30] F19 product type [8], [25], [29], F20 product-description length/words [4], F21 product feature [27], F22 product controls [29], F23 monthly controls [29], F24 product quality [29], F25 reviewer ranking [1], F26 reviewer reputation [1], [27], F27 reviewer expertise [7], [29], F28 reviewer non-anonymity [29] F29 reviewer social interaction and profile [27], F30 first-person singular pronouns (FPSP) affect [28], F31 trustworthiness [7]	Amazon.com 2014 [1], [7]–[9], [24], [25], [27]–[30] Amazon.cn [26]	10-fold cross validation (CV) [9]	M1 Tobit regression [7], [8], [25], [27]–[29] M2 linear regression [24], M3 linear-support vector regression [24], M4 regression [26], M5 regression in CNN/MSE Loss [9], M6 support vector regression [24], [26], M7 support vector machine—radial basis function [27], M8 negative binomial regression [1], M9 random forest [1] M10 ensemble model (top layer: random forest, lower layer: ordinary least square regression, linear ridge regression and gradient boosted machine) [30] M11 logistic regression [4]
R3 helpful vote [2], [4], [5], [31], [32] R4 categorized helpful vote [3], [19], [33], [34]	F1 [2]–[4], [19], [31], F2 [2], [4], [5], [19], [32], F3 [19], F4 [3], [19], [31], [34], F5 [3], [19], F6 [4], [32], F7 , F9 , F11 , F15 , F20 [4] , F25 [5], F27 , F31 [7] F41 compound sentiment [32], F42 review age/weeks [31], F43 review age/years [2], F44 Amazon service [2], F45 subjective expression [2], F46 flavor feature [2], F47 number of images [3], [33], F48 existence of images [19], F49 review informativeness [3], F50 readability/Simple Measure of Gobbledygook [19], F51 readability/Flesch Reading Ease Index [5], F52 syllable [19], F53 reviewer-related (anonymity, user image, membership, user experience value) [3], [33] F54 verified purchase [19]	Amazon.com 2014 [2], [4], [5], [19], [32], JD.com [3], TripAdvisor [33], Ciao.co.uk [34], Drugs.com, Yelpf.com [5]	10-fold CV [3], Time-based [19], Total sampling [19], Simple random [19]	M2 [4], M4 [5], [32], M8 [2], M9 [3], M11 [33], M11 logistic regression in XGB classifier [19] and CNN/cross entropy loss [19], [34] M12 Zero Inflated Poisson [31]
This study helpful vote (R3)	review text (F4)	Amazon.com 2018 [22] and IMDb [23]	Adjusted 10-fold CV (ACV), adaptive window size (AWS)	Distribution-adapted model in LM, XGB and CNN

helpfulness rating, which comes from the ratio of the number of helpful votes to the total votes. Tobit modeling, a zero-censored regression, revises the regression model on massive zero-value problems and has become a popular model for predicting the helpfulness rating [7], [8], [25], [27], [28], [29]. When implemented in machine learning, the regression and Tobit modeling employ the same objective functions, Sum Squared Error (SSE) or MSE. Those objective functions come from the initial assumption that the dependent variable in the model is normally distributed [12].

Researchers were motivated by the results in the helpfulness rating to continue to employ both models to estimate the number of helpful votes [4], [6], [7], [32] although the distribution is not normal. The central limit theory also supports this condition that a large dataset tends toward a normal distribution in many situations, even if the original variables themselves are not normally distributed [38].

Recently, considering the discrete form of helpful votes, negative binomial regression has been on the rise as a popular model for predicting the number of helpful votes [1], [2], [3]. This model assumes that the number of helpful votes is in a discrete distribution with an over-dispersion problem [1], [2], [3], where the variance is far from the mean value [39].

C. FACTORS IN HELPFUL VOTE PREDICTION MODELS

Previous research has generally used numerical factors to estimate the helpfulness of the reviews [1], [2], [3], [4], [7], [8], [19], [25], [26], [27], [28], [29]. Numerical factors, such as star rating [1], [2], [3], [4], [7], [8], [19], [25], [26], [28], [29], [32], review age [1], [2], [3], [19], [29], product type [8], [25], [29], and number/existence of product images [3], [19], have been proven to be known as factors that can significantly have impacts on estimating the helpfulness.

In addition, review length [1], [2], [4], [7], [8], [25], [26], [27], [28], [29], [32], readability [4], [19], [27], [29], review sentiment [1], [29], and text complexity [1], which are generated from the text reviews, are also essential in predicting the helpfulness rating. Some studies prove that the review helpfulness is significantly influenced by the contents of a review, which describe other customers' experiences or emotions [25].

However, recent studies focus on the review contents represented by word embedding [9], [19], [40] or bag-of-words vector [19], [24]. Moreover, a model with mixed numerical and text factors gives no significant improvement compared to models using either text or numerical factors [19].

D. SAMPLING METHODS

Helpfulness prediction studies generally use random sampling methods on the Amazon dataset. This method randomly chooses elements of the training and testing data. The 10-fold cross-validation sampling method is one of the most popular random sampling methods [3], [9]. However, the random sampling-based models do not consider review age, which neglects the obsolescence of the product functions or features in the reviews [19].

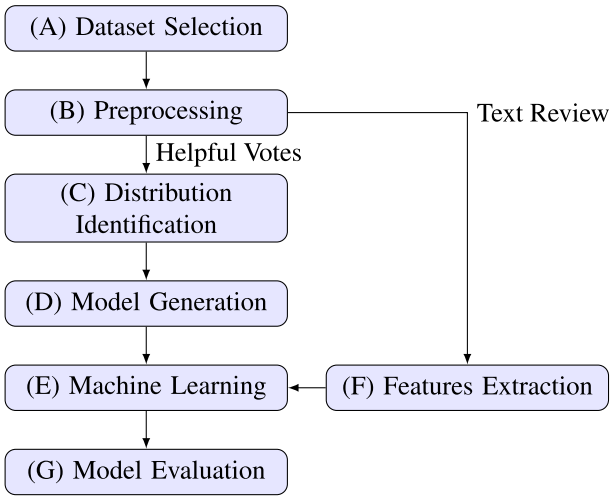


FIGURE 1. Flowchart illustrating the steps in our proposed framework.

Considering review age, Saptono and Mine [19] proposed TBS methods. Their methods use Cochran's formula and time range to calculate the adequate training set size in classification tasks.

III. PROBABILITY DISTRIBUTION FUNCTION

This study assumes that the helpful vote y is in the continuous EDM family. The native members of EDM family are the Normal, Gamma, and InvGauss distributions [11]. Regarding the central limit theorem [38], of these distributions, distributional approximation for a wide range of data tend to approach a normal distribution. Therefore, we add Expon and Wald, as the particular case of Gamma and InvGauss, respectively. Both are also EDM family members.

Each distribution in the EDM family has a different probability density function (PDF). However, we can generate a common structure of the distribution PDF $f(y, \theta, \phi)$ from the response variable y, with parameters θ and ϕ, as follows:

$$f(y, \theta, \phi) = a(y, \phi) \exp\left\{\frac{y\theta - \kappa(\theta)}{\phi}\right\}, \tag{1}$$

where θ is called the *canonical function*, $\kappa(\theta)$ is called the *cumulant function*, ϕ is the dispersion parameter and $a(y, \phi)$ is a normalizing function ensuring that (1) is a probability function [11]. We employ (1) to identify the distribution of helpful votes and develop the models.

The distribution-adapted model is generalized from linear regression analysis, the normal distribution-adapted model. Therefore, we use the mean symbol μ of the normal distribution to represent the estimator $E[y]$ for the variable y in all distributions.

IV. PROPOSED FRAMEWORK

In this section, we elaborate on our proposed framework. Fig. 1 shows the overall steps in our framework. We first select the review dataset in Step A and preprocess it in Step B. From Step B, we choose the helpful votes as the dependent

variable and the text part of reviews as the independent variable. Subsequently, we identify the distribution of helpful votes in Step C. In Step D, a generalized linear model is formulated based on the distribution of helpful votes. Next, in Step E, we implement the distribution-adapted models in machine learning. Text factors extracted from a review dataset take two forms: bag-of-words and word-embedding, which follow machine learning, in Step F. Finally, in Step G, the models are evaluated on the dataset using adaptive window sampling methods and measuring the performance by the MAE metric.

A. DATASET SELECTION AND PREPROCESSING

This study uses three categories of Amazon dataset [22] for Step:

1) Automotive (AD1),
2) Cell Phones and Accessories (AD2), and
3) Industrial and Scientific (AD3).

Those datasets are a combination of many products, each of which has the same category.

We also use movies of IMDb dataset [23]. We select three movies dataset as follows:

1) La La Land 2016 (ID1),
2) X-Men Apocalypse 2016 (ID2),
3) 3 Idiots 2009 (ID3).

Amazon datasets in Table 2 contain massive numbers of inapplicable votes, and then in Step B, we apply three rules to select the data that are involved in the experiments. First, we only use non-zero/applicable vote reviews in the experiments because it is unclear whether an inapplicable vote review is new or unhelpful. Even though the position of the zero votes reviews is in the middle of voted reviews, it could be a 'never seen' review due to the system design which gives a priority to popular reviews. Second, we drop duplicate reviews and leave the original one in the dataset. The removed duplicate comes from the system that shares one review for items with variations, such as color and size. Each variation has a unique identity number but shares the same reviews, making the duplicate review not directly related to the item. That is why the duplicate reviews more frequently appear on Amazon datasets than on IMDb datasets, as shown in Table 2. Third, we apply L2-normalization [41] to the number of helpful votes. The big difference between mean and variance in Table 2 shows that the over-dispersion problem occurs in the helpful votes of all datasets.

TABLE 2. Dataset description.

Dataset	Number of reviews	Applicable	Unique (N)	Mean	Variance
Amazon					
AD1	1,711,519	190,796	180,001	5.90	487.92
AD2	1,128,437	92,001	88,804	9.38	1156.78
AD3	77,071	9,620	8,878	9.92	1881.03
IMDb					
ID1	1,735	1,574	1,573	13.16	2691.82
ID2	830	701	701	13.59	1462.31
ID3	600	357	356	14.08	1533.26

From Step B, we select the helpful votes as the dependent variable and the text reviews as the independent variable. We feed the helpful votes to Step C and the text reviews to Step F.

B. DISTRIBUTION IDENTIFICATION

In Step C, we employ the MSE and AIC scores to determine the goodness of fit [13], [14] to identify the distribution of helpful votes. We compared those scores among five distributions in EDM $\mathcal{C}$: Normal, Gamma, InvGauss, Expon, and Wald. We initially generate a histogram based on the whole helpful votes of each dataset to obtain MSE and AIC. We employ Scott's rule [15] to find the number of histogram bins. This rule considers the data characteristics and the size of data in the number of bin formulations, as shown in (2).

The steps of obtaining the MSE and AIC scores for each distribution in $\mathcal{C}$ are described as follows:

1) We first select a distribution c in $\mathcal{C}$ and fit it to the helpful votes of the dataset to get parameters of c.
2) We generate a histogram of the helpful votes of the dataset. The number of bins n_b in the histogram is calculated using Scott's rule [15] in (2).
$$n_b = \frac{max - min}{3.49\sigma N^{-\frac{1}{3}}}, \tag{2}$$
where σ, N, max, and min are standard deviation, the dataset size, the maximum and the minimum value of the helpful votes, respectively. In this step, we also get n_b of (x_i, y_i) for each bar in the histogram, where y_i represents the actual value of helpful votes in the axis x_i.
3) Based on n_b calculated in step-2 and the parameters obtained in step-1, we generate n_b of $\hat{y}_i$ by using the density function of c.
4) MSE and AIC are calculated using y_i from step-2 and generated data, $\hat{y}_i$ in step-3
$$MSE = \frac{\sum_{i=1}^{nb}(\hat{y}_i - y_i)^2}{n_b - k} \tag{3}$$
$$AIC = 2k - 2\max_i \log \hat{y}_i, \tag{4}$$
where k is the number of parameters in the distribution c.
5) The steps above are repeated for other distributions in $\mathcal{C}$.

The distribution with the least MSE and AIC scores is the best approximate distribution. We calibrate those results with a KS score obtained by the KS test output, as (5):

$$D_{m,n} = \sup_x |F_{1,m}(x) - F_{2,n}(x)|, \tag{5}$$

where $D_{m,n}$ is a KS score for two sample with size n and m, sup is the supremum function, $F_{1,m}$ and $F_{2,n}$ are empirical cumulative distribution functions from sample 1 and sample 2, respectively. We also use this calibration to answer Q1 in Section I.

TABLE 3. Unit deviance of EDM $d(y, \mu)$ [11] with canonical function (θ), dispersion parameter (ϕ), cumulant function ($\kappa(\theta)$), estimator of y (μ) and variance of y (σ^2).

Distribution	θ	ϕ	$\kappa(\theta)$	$d(y, \mu)$
Normal	μ	σ^2	$\frac{\theta^2}{2}$	$(y-\mu)^2$
Gamma	$-\frac{1}{\mu}$	ϕ	$-\log(-\theta)$	$2\{-\log\frac{y}{\mu}+\frac{y-\mu}{\mu}\}$
InvGauss	$-\frac{1}{2\mu^2}$	ϕ	$-\sqrt{-2\theta}$	$\frac{(y-\mu)^2}{\mu^2 y}$

C. MODEL GENERATION

The main task of this study is to generate a model that adapts the suitable distribution of helpful votes in Step. The critical process is to generate the unit deviance for the objective function. The deviance is a generalization of using SSE in regression analysis, which also plays a role as a cost function and has to be minimized [12]. Because μ estimator $E[y]$ and θ in (1) are a one-to-one function [11], then we get the formula of unit deviance d for response y and the estimator μ is as follows:

$$d(y, \mu) = 2(t(y, y) - t(y, \mu)), \tag{6}$$

where $t(y, \mu)$ is the order of exponential in (1), which is defined as follows:

$$t(y, \mu) = y\theta - \kappa(\theta), \tag{7}$$

where θ is a function of μ.

Generalizing linear regression, we get the deviance as the summation of unit deviance in (6). The unit deviance of the distribution used in this paper is shown in Table 3.

Expon is a particular form of Gamma distribution with shape parameter equal to one and scale parameter θ, so the PDF of the Expon distribution $f(y, \theta)$ is shown in (8) [11].

$$f(y, \theta) = \exp\left\{y\left(\frac{-1}{\theta}\right) - \log\theta\right\} \tag{8}$$

Based on (1), (6), (7) and $E(y) = \mu = \theta$, we get

$$t(y, \mu) = y(-\frac{1}{\mu}) - \log\mu \tag{9}$$

and the unit deviance of the Expon distribution is as follows:

$$d(y, \mu) = 2\left\{-\log\frac{y}{\mu} + \frac{y-\mu}{\mu}\right\}. \tag{10}$$

We found the equivalence of the Expon unit deviance shown in (10) with the parent distribution, Gamma, as shown in Table 3.

On the other hand, Wald is a particular case of InvGauss with μ as the estimator of y is equal to one. Generalizing Expon, we generate the unit deviance of Wald, which is equivalent to the parent distribution, InvGauss, as shown in Table 3. The unit deviance functions instead of the normal deviance have μ for the denominator. This condition will affect the result if the prediction is close or equal to zero. Therefore, we need to apply a translation to deviance.

For the normal deviance $d(y, \mu)$, it applies deviance translation as follows:

$$d(y + \epsilon, \mu + \epsilon) = d(y, \mu), \tag{11}$$

where μ is an estimator for the response variable y and ϵ is the translation coefficient. A typical example of the deviance translation is implemented in squared log error (SLE), with ϵ equal to one. Based on the deviance translation for the normal distribution and SLE, we generalize deviance translation for other deviance to prevent error division by zero or an anomaly result by a number close to zero.

D. MACHINE LEARNING AND FEATURE EXTRACTION

We employ three types of machine learning in Step. First, considering the widespread use of regression and Tobit model in previous studies, we employ the linear model (LM). Second, we employ XGBoost (XGB) [20] since it has an extraordinary result on classifier task as mentioned in [19]. Finally, regarding a state-of-the-art helpfulness rating prediction model [9], we employ CNN to implement models adapting to the distribution of helpful votes. Furthermore, we use the unit deviance, the output of Step, as the objective function in machine learning to be minimized. For Gamma and Expon distributions, we develop only the best one according to the distribution identified. This condition is also applied to the InvGauss and Wald distributions.

XGB employs gradient boosting and Taylor expansion to support the development of a custom objective function. Therefore, we need to provide the first and second derivatives from each deviance [20] in Table 3.

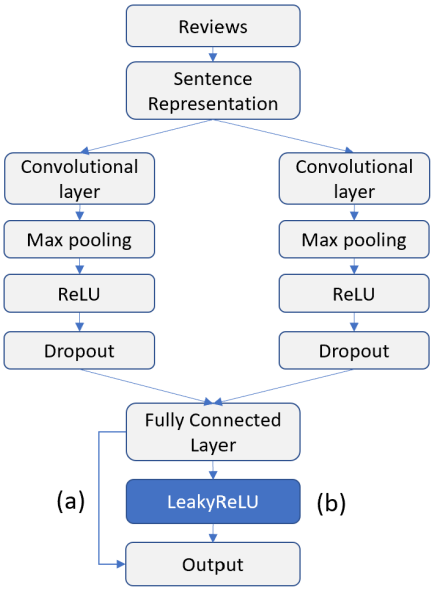


FIGURE 2. CNN architecture for models adapting to (a) normal distribution (linear regression) in [9] (b) normal (Tobit), Gamma, InvGauss, Expon and Wald distributions.

We employ the CNN based on the architecture proposed in [9], which is a state-of-the-art helpfulness rating prediction

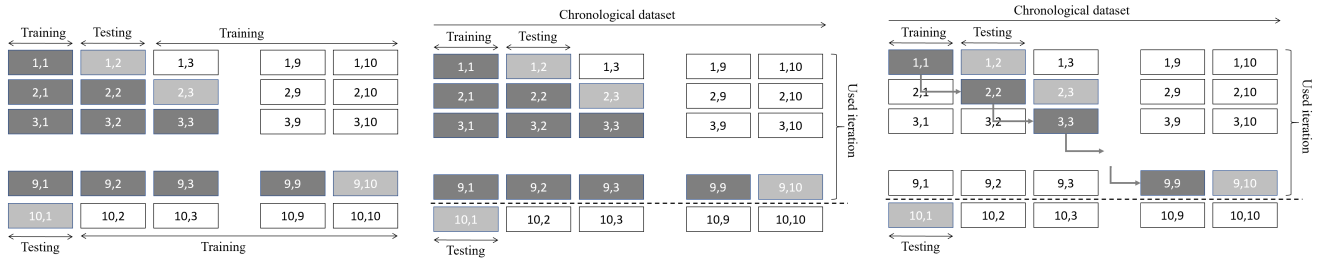


FIGURE 3. Sampling methods abstraction for (a) the 10-folds cross validation (CV) on random dataset, (b) the adjusted 10-folds cross-validation (ACV) on the dataset sorted in chronological order, and (c) adaptive window size(AWS) on the dataset sorted in chronological order. Our proposed sampling method AWS adjust the size of training size of TBS [19].

model, with modifications of the loss function and output layer, as shown in Fig. 2. We build the loss function based on the deviance of the distribution, which is the summation of the unit deviance shown in Table 3. When implementing Tobit model and models adapting to Gamma/Expon, and InvGauss/Wald distributions in CNN, we employ an output layer to prevent the result from negative value. Two activation functions: ReLU [42] and LeakyReLU [43] with negative coefficient, a, as in (12) are possible to use. Since ReLU has a problem called the 'dying' phenomenon, where the prediction is always zero for every value in the dependent variable, we use LeakyReLU. Later, we prove the dying phenomenon when using ReLU for the activation function at the output layer.

$$f(x) = max(ax, x). \tag{12}$$

In LM, we employ the same objective function as in CNN. We use the Limited Memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) approach [44] to minimize the objective function.

As independent variables, we use the results of feature extraction from the text review, in Step, depending on machine learning. In the LM and XGB-based models, the text parts of the reviews are transformed into unigrams, and bigrams term frequency-inverse document frequency (TF-IDF) [45] weighted bag-of-words format as independent variables. Meanwhile, in CNN, the text parts are transformed into word embedding by applying GloVe [46], [47] with six billion words and 100 dimension vectors (Glove.6B.100d).

E. MODEL EVALUATION

Finally, in Step G, to evaluate the model, we propose an adaptive window size (AWS) sampling method. AWS is inspired by the TBS method [19], as shown in Fig. 3(b). The basic idea of the TBS method is to get the training set as close as possible to the testing set, under the assumption that, as the training set gets closer to the testing set, it shares a more similar characteristic with the testing set, and the model performance becomes improved [19].

AWS uses a variable length of training set instead of a fixed-length training set. Since the testing set is the same, we can select a training set suitable to the testing set. However, Cochran's formula uses the variance of the binomial distribution to determine the sample size [48]. According to the central limit theorem, we need to deal with continuous distributions. Therefore, we adjust the formula as in (13) to get the sample size n from the dataset size N.

$$n_0 = \frac{4Z^2\sigma^2}{w^2} \quad \text{and} \quad n = \frac{n_0}{1 + \frac{n_0 - 1}{N}}, \tag{13}$$

where Z is a standard score for the desired confidence level, σ^2 is the variance of helpful votes, w is a unit margin of error, and n_0 is the number of samples if the dataset size is unknown. In this paper, we calculate σ from the helpful votes of reviews in the training set because we assume that reviews after the training set are not voted yet.

If the dataset size is N, and fold-size is f, then AWS is as follows:

1) We first sort the dataset with size N in chronological order and divide the dataset by fold-size f. We use the first data with size $N_t = N/f$ for the candidate training set and the next dataset with N_t for the testing set.
2) We calculate the variance of the helpful votes σ_t^2 in the candidate training set in step (1). Furthermore, we fed σ_t^2 and the size of the candidate training set size N_t to Cochran's formula in (13), replacing σ^2 and N to get the sample size n.
3) If n in step (2) $\geq N_t$ in step (1), then we use N_t as the training set. Otherwise, we use n elements in the candidate training set closest to the testing set.
4) We add the testing set to the candidate training set and select the following N_t data as the new testing set. We feed the new candidate training set and testing set to step (2).
5) We repeat the above for $f - 1$ samples.

For comparison, we run the model in 10-fold cross-validation (CV). Meanwhile, we need to adjust the 10-fold CV in Fig. 3(a) when applying the model to the dataset in chronological order, as shown in Fig. 3(b). The white cells are used as a training set in random sampling. Since the review dataset is sorted in chronological order, the white cells are unlabeled. The testing set in the current row increments the

TABLE 4. Distribution-adapted and baseline models, including some miscellaneous linear and ensemble models.

Machine Learning	Model	Objective Function	Distribution Assumption	Abbreviation
Misc.	Stochastic Gradient Descent	Mean Squared Error (MSE)	Normal	SGD
	Multi Task Elastic-Net	Sum Squared Error (SSE)*	Normal	NET
	Orthogonal Matching Pursuit	SSE*	Normal	OMP
	Least Angle Regression	SSE*	Normal	LAR
	LASSO	SSE*	Normal	LAS
	LAR with LASSO Regularization	SSE*	Normal	LAL
	Bayesian Automatic Relevance Determination	SSE*	Normal	ARD
	Random Forest Regression	MSE	Normal	RFR
	Gradient Boosting Regression	MSE	Normal	GBR
	Extra Tree Regression	MSE	Normal	ETR
LM	Linear Regression [4]	MSE	Normal	LSE
	Tobit Regression [6], [7]	MSE	Normal	LTR
	Gamma/Expon	Gamma/Expon Deviance	Gamma/Expon	LEX
	InvGauss/Wald	InvGauss/Wald Deviance	InvGauss/Wald	LWA
XGB	Linear Regression	MSE	Normal	XSE
	Tobit Regression	MSE	Normal	XTR
	Gamma/Expon	Gamma/Expon Deviance	Gamma/Expon	XEX
	InvGauss/Wald	InvGauss/Wald Deviance	InvGauss/Wald	XWA
CNN	Linear Regression [9]	MSE	Normal	CSE
	Tobit Regression	MSE	Normal	CTR
	Gamma/Expon	Gamma/Expon Deviance	Gamma/Expon	CEX
	InvGauss/Wald	InvGauss/Wald Deviance	InvGauss/Wald	CWA

Note: ** : with a modification as an effect of regularization.*

training set in the next row, and we call it the adjusted cross-validation (ACV) sampling method.

We employ mean absolute error (MAE) in (14), to evaluate the performance of each model and compare it among the models to find the best model. The model with smaller MAE means better model.

$$MAE = \frac{1}{n}\sum_{i=0}^{n-1}|y_i - \hat{y}_i|, \tag{14}$$

where $y, \hat{y}$ are the actual and predicted numbers of helpful votes, respectively, and n is the testing set size.

On the other hand, we need to use mean absolute percentage error (MAPE), as shown in (15), to prove the dying phenomenon of CNN with ReLU on the output layer. Using MAPE makes it easier to detect zero prediction on any values of normalized helpful votes. If the average MAPE is 1 with 0 standard deviations, we can conclude that the prediction value is always zero.

$$MAPE = \frac{1}{n}\sum_{i=0}^{n-1}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{15}$$

We also use MAE for CTR with ReLU on the output layer in determining the acceptance of a model performance. If the model has a smaller MAE than the threshold, it has an acceptable result, otherwise is an unacceptable one.

V. PERFORMANCE EVALUATION

We conduct experiments to validate our proposed models. We first identify the distribution using MSE and AIC scores. Subsequently, we implement the distribution-adapted models using three machine learning methods: LM, XGB, and CNN, and evaluate them using the AWS and ACV sampling methods. We employ a statistical analysis of variance to check the significance of the mean difference. Finally, we investigate the effect of machine learning, sampling methods, and distribution on model performance.

A. EXPERIMENT SETUP

We build two baseline models adapted to the normal distribution: linear regression [4] and Tobit regression [6], [7] when the best approximation distribution is not the normal distribution. In addition, we also involve a model adapting to the second-best approximate distribution to investigate the possibility of model performance following the distribution of helpful votes. We develop each model in three machine learning contexts, where the details are shown in Table 4.

Here, we implement our proposed framework in `Python`. We use a linear model library on `sklearn.linear_model` to implement linear regression, Tobit regression, and the Gamma/Expon distribution-adapted model. For the InvGauss/Wald distribution-adapted model, we use a Tweedie Regression and set the power with three. Since we use a linear model on `sklearn.linear_model`, we also use all machine learning in the library and ensemble, as shown in Table 4, which employ SSE/MSE or its modification for the objective function as baseline models [49], [50], [51], [52], [53], [54], [55], [56].

We also use `xgboost` library to implement XGB. Two problems arise when we use XGB:

1) The native objective function of Gamma/Expon can not handle a normalized value of the independent variable,
2) The objective function of InvGauss/Wald has not been implemented in XGB yet, while Tweedie regression in XGB can not accept power equal to or close to three.

Therefore, we need to provide a custom objective function for models adapting to Gamma/Expon and InvGauss/Wald

distributions. We first use a translation of unit deviance in Table 3, as generalization from (11). Equations (16) and (17) show the translation of the Gamma/Expon and InvGauss/Wald unit deviance, which we call GAMMA+ϵ and WALD+ϵ, respectively. Subsequently, we feed the first and second derivatives of the translation of unit deviance to develop custom objective functions of XGB.

$$d(y+\epsilon,\mu+\epsilon) = 2\Big\{-\log\frac{y+\epsilon}{\mu+\epsilon}+\frac{y-\mu}{\mu+\epsilon}\Big\} \quad (16)$$

$$d(y+\epsilon,\mu+\epsilon) = \frac{(y-\mu)^2}{(\mu+\epsilon)^2(y+\epsilon)} \quad (17)$$

Appendix A Figs. 12 and 13 show that the GAMMA+ handles normalized helpful votes better than the original Gamma objective function in XGB. GAMMA+ performs better when $\epsilon \geq 1$. Identically, the WALD+ implementation in XGB also handles the normalized helpful votes better than the original Wald objective function, as shown in Figs. 14 and 15 in Appendix A. WALD+ performs better when $\epsilon \geq 1$ with AWS on ID3 and $\epsilon \geq 2$ for the rest. We resume the ϵ-values in Table 5.

Moreover, we use `PyTorch` to develop CNN. We use `MSE Loss` for models adapted to a normal distribution. However, we need to provide a custom loss function for models adapting to Gamma/Expon and InvGauss/Wald distributions based on the mean of summation from unit deviance, as shown in Table 3.

We can use two activation functions: ReLU and negative coefficient LeakyReLU, as mentioned in Subsection IV-D, for the output layer of CTR, CEX, and CWA. Since CTR is one of the baseline models, we use the CTR model to prove that CTR with ReLU for the output layer will give the 'dying' phenomenon on normalized helpful votes. On the other hand, CNN with a negative coefficient LeakyReLU will solve the ReLU problem.

To prove the 'dying' phenomenon, we combine CTR with ReLU for the output layer. CTR model with ReLU gives the average of MAPE equal to 1 with 0 standard deviations for all datasets with both sampling methods, ACV and AWS. This result proves that the combination of CTR and ReLU always gives zero results for any value of the normalized helpful votes. Since the output is always zero, we get the average MAE of CTR-ReLU as the average sum squared of the absolute actual number of helpful votes. So, it is unacceptable if the value of MAE of any model is greater than or equal to the average of MAE of CTR-ReLU. Therefore, we use the average MAE of CTR-ReLU as a threshold to determine the acceptance of the model performance.

Furthermore, considering the value of the helpful votes after L2-normalization, we use negative values for the LeakyReLU coefficient in the range $[-1e^{-3}, -1e^{-9}]$ for the output layer of CTR to solve the ReLU problem. The negative coefficient of LeakyReLU, as in (12), ensures the output is above the axis line ($y = 0$), except on 0. In addition, within that range, we also get a gentle slope of LeakyReLU. Fig. 16 in Appendix shows that CTR with a negative coefficient LeakyReLU has an improvement compared to CTR with ReLU. The LeakyReLU coefficient for each dataset is shown in Table 5.

TABLE 5. The ϵ-values for GAMMA+ϵ and WALD+ϵ, and the coefficients for CNN-based models.

Dataset	GAMMA+ϵ		WALD+ϵ		CTR - LeakyReLU	
	ACV	AWS	ACV	AWS	ACV	AWS
Amazon						
AD1	2	5	5	5	-1.0E-7	-1.0E-5
AD2	4	5	5	5	-1.0E-8	-1.0E-5
AD3	5	5	4	5	-1.0E-5	-1.0E-6
IMDb						
ID1	5	5	5	5	-1.0E-5	-1.0E-5
ID2	5	5	4	4	-1.0E-4	-1.0E-4
ID3	5	5	4	4	-1.0E-4	-1.0E-4

TABLE 6. Distribution identification results. We use two goodness of fit metrics: AIC and MSE. We also provide the KS score for calibration.

Goodness of fit	Distribution	Dataset					
		AD1	AD2	AD3	ID1	ID2	ID3
	n_b	4286	766	319	52	38	19
MSE	Normal	4.24E-5	7.06E-5	3.52E-5	5.51E-5	6.65E-5	8.40E-5
	Gamma	4.70E-5	7.65E-5	4.11E-5	6.97E-5	9.39E-5	3.40E-5
	InvGauss	4.71E-5	7.90E-5	4.12E-5	2.85E-5	2.16E-5	9.41E-5
	Expon	1.32E-5	2.42E-5	5.33E-6	6.53E-6	7.30E-6	**6.07E-6**
	Wald	**7.12E-6**	**1.08E-5**	**2.89E-6**	**5.69E-6**	**4.61E-6**	1.07E-5
AIC	Normal	12.028	12.892	13.382	13.743	13.142	13.180
	Gamma	20.207	15.003	21.347	22.572	22.459	11.565
	InvGauss	25.678	22.838	24.724	11.692	11.002	13.956
	Expon	5.071	6.357	7.061	**8.255**	8.247	**8.546**
	Wald	**4.588**	**5.749**	**6.940**	8.322	**8.185**	8.824
KS	Normal	0.432	0.413	0.424	0.409	0.381	0.360
	Gamma	0.989	0.634	0.780	0.806	0.839	0.306
	InvGauss	0.604	0.639	0.654	**0.182**	**0.106**	0.357
	Expon	0.396	0.409	0.406	0.444	0.285	0.399
	Wald	**0.296**	**0.267**	**0.266**	0.276	0.121	**0.228**

Note: *The boldface is the best approximate distribution of each goodness of fit.*

We need to provide some parameters in machine learning and the AWS sampling method. We use a learning rate of 0.1 for XGB and 0.01 for CNN. We also set a negative parameter of 1e-5 to get a positive gradient function on LeakyReLU. In the AWS sampling method, we set the unit margin of error w as two and Z-score with a confidence level of 99%.

B. APPROXIMATE DISTRIBUTION

Since there is no best fit distribution based on the p-value of the KS test with a value less than 0.01 for all distributions, we use MSE and AIC scores to establish the approximate distribution to identify the distribution of helpful votes. Table 6 shows the result of approximate distribution identification using the MSE and AIC scores compared to the KS score as calibrator.

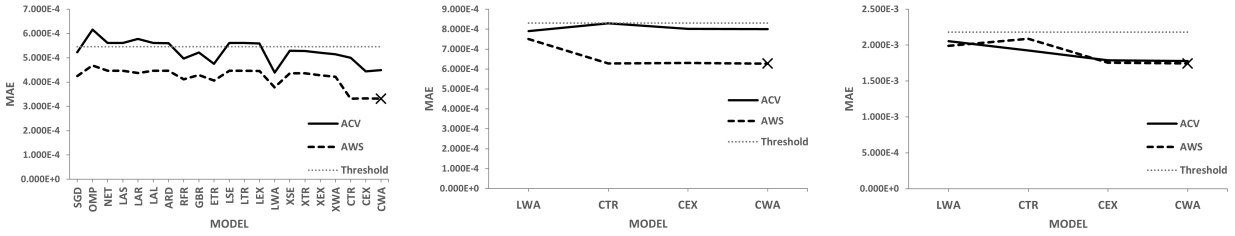
We get Wald as the best approximate distribution for Amazon datasets, either with MSE or AIC. This result is the same as the outcome of the KS score best approximate distribution, as shown in Table 6. Table 6 shows the different results for IMDb datasets, although the score difference is small between Expon and Wald with AIC.

Above results answer Q1 that our approach gives the same result on the best approximate distribution as the KS score on Amazon datasets. However, we get dynamic results on small and homogenous datasets: IMDb. MSE gives Wald on ID1 and ID2 datasets and Expon on ID3. Meanwhile, AIC provides Expon on ID1 and ID3 datasets and Wald on ID2. Those

TABLE 7. Performance of models on Amazon datasets with the ACV and AWS sampling methods. The performance is measured by the average of MAE followed by the standard deviation in the parentheses. We use the result of CTR-ReLU as the acceptance threshold of models.

Model	Amazon Dataset					
	AD1		AD2		AD3	
	ACV	AWS	ACV	AWS	ACV	AWS
Threshold	5.461E-4 (8.42E-5)	5.461E-4 (8.42E-5)	8.309E-4 (1.48E-4)	8.309E-4 (1.48E-4)	2.181E-3 (6.60E-4)	2.181E-3 (6.60E-4)
Misc.						
SGD	**5.225E-4** (1.28E-4)	**4.245E-4** (1.18E-4)	9.367E-4 (1.72E-4)	9.314E-4 (1.91E-4)	2.921E-3 (7.29E-4)	2.863E-3 (6.51E-4)
OMP	6.165E-4 (2.36E-4)	**4.684E-4** (1.12E-4)	1.052E-3 (2.40E-4)	1.031E-3 (2.56E-4)	4.573E-3 (1.31E-3)	4.588E-3 (1.33E-3)
NET	5.612E-4 (1.45E-4)	**4.464E-4** (1.33E-4)	1.012E-3 (1.80E-4)	9.295E-4 (2.03E-4)	2.944E-3 (6.92E-4)	2.814E-3 (7.21E-4)
LAS	5.612E-4 (1.45E-4)	**4.464E-4** (1.33E-4)	1.012E-3 (1.80E-4)	9.295E-4 (2.03E-4)	2.944E-3 (6.92E-4)	2.814E-3 (7.21E-4)
LAR	5.777E-4 (2.16E-4)	**4.375E-4** (1.15E-4)	1.028E-3 (2.58E-4)	9.920E-4 (2.51E-4)	5.455E-3 (3.27E-3)	4.798E-3 (1.89E-3)
LAL	5.612E-4 (1.45E-4)	**4.464E-4** (1.33E-4)	1.012E-3 (1.80E-4)	9.295E-4 (2.03E-4)	2.944E-3 (6.92E-4)	2.814E-3 (7.21E-4)
ARD	5.602E-4 (1.43E-4)	**4.464E-4** (1.33E-4)	9.977E-4 (1.75E-4)	9.189E-4 (1.98E-4)	3.759E-3 (1.28E-3)	3.719E-3 (1.31E-3)
RFR	**4.969E-4** (1.12E-4)	**4.113E-4** (1.07E-4)	9.258E-4 (1.70E-4)	8.851E-4 (1.84E-4)	2.735E-3 (7.91E-4)	2.664E-3 (8.20E-4)
GBR	**5.217E-4** (1.29E-4)	**4.288E-4** (1.12E-4)	9.241E-4 (1.76E-4)	8.845E-4 (1.72E-4)	2.856E-3 (7.18E-4)	2.721E-3 (7.27E-4)
ETR	**4.746E-4** (1.11E-4)	**4.068E-4** (9.99E-5)	9.023E-4 (1.77E-4)	8.618E-4 (1.84E-4)	2.713E-3 (5.82E-4)	2.640E-3 (6.57E-4)
LM						
LSE	5.612E-4 (1.45E-4)	**4.464E-4** (1.33E-4)	1.012E-3 (1.80E-4)	9.295E-4 (2.03E-4)	2.944E-3 (6.92E-4)	2.815E-3 (7.21E-4)
LTR	5.612E-4 (1.45E-4)	**4.464E-4** (1.33E-4)	1.012E-3 (1.80E-4)	9.295E-4 (2.03E-4)	2.944E-3 (6.92E-4)	2.815E-3 (7.21E-4)
LEX	5.595E-4 (1.44E-4)	**4.455E-4** (1.33E-4)	1.005E-3 (1.78E-4)	9.245E-4 (2.00E-4)	2.925E-3 (6.89E-4)	2.793E-3 (7.18E-4)
LWA	**4.389E-4** (8.23E-5)	**3.788E-4** (9.26E-5)^{c}	**7.906E-4** (1.23E-4)	**7.510E-4** (1.34E-4)	**2.053E-3** (5.41E-4)	**1.991E-3** (5.66E-4)
XGB						
XSE	**5.288E-4** (1.22E-4)	**4.360E-4** (1.05E-4)	9.340E-4 (1.74E-4)	8.974E-4 (1.88E-4)	2.846E-3 (7.44E-4)	2.777E-3 (7.70E-4)
XTR	**5.281E-4** (1.21E-4)	**4.358E-4** (1.05E-4)	9.338E-4 (1.74E-4)	8.965E-4 (1.88E-4)	2.833E-3 (7.45E-4)	2.772E-3 (7.71E-4)
XEX	**5.206E-4** (1.21E-4)	**4.279E-4** (1.06E-4)	9.105E-4 (1.70E-4)	8.707E-4 (1.70E-4)	2.802E-3 (6.51E-4)	2.751E-3 (6.59E-4)
XWA	**5.141E-4** (1.15E-4)	**4.215E-4** (1.04E-4)	9.004E-4 (1.59E-4)	8.518E-4 (1.66E-4)	2.755E-3 (6.33E-4)	2.655E-3 (6.54E-4)
CNN						
CSE	1.045E-3 (1.17E-3)	3.523E-3 (2.78E-3)	2.480E-3 (3.55E-3)	3.464E-3 (3.00E-3)	1.121E-2 (3.79E-3)	1.349E-2 (1.03E-2)
CTR	**4.999E-4** (8.70E-5)	**3.322E-4** (7.43E-5)^{b}	**8.299E-4** (1.49E-4)	**6.283E-4** (1.29E-4)^{b}	**1.926E-3** (7.00E-4)	**2.088E-3** (6.19E-4)
CEX	**4.438E-4** (8.44E-5)	**3.325E-4** (9.60E-5)^{b}	**8.012E-4** (1.06E-4)	**6.306E-4** (1.31E-4)^{b}	**1.790E-3** (5.48E-4)^{c}	**1.754E-3** (5.88E-4)^{b}
CWA	**4.492E-4** (8.54E-5)	**3.315E-4** (7.64E-5)^{a}	**7.999E-4** (1.07E-4)	**6.273E-4** (1.35E-4)^{b}	**1.778E-3** (6.06E-4)^{c}	**1.743E-3** (6.36E-4)^{a}

Note: *The boldface means the model has an acceptable result. Results with* ^{a} *are significant for* $p \leq 0.01$, ^{b} *for* $p \leq 0.05$ *and* ^{c} *for* $p \leq 0.1$.



(a) AD1 (b) AD2 (c) AD3

FIGURE 4. Performance of models under the threshold line on Amazon datasets. Distribution-adapted models in LM and CNN do not always cross the line.

results differ from the results of the KS score, which gives InvGauss on ID1 and ID2, and Wald on ID3. Furthermore, we feed the results of distribution identification to the model generation step.

According to the results in Table 6, the best approximate distribution on Amazon datasets is Wald, whose effect may depend on machine learning methods. We also implement Expon, the second-best approximate distribution, instead of the parent Gamma to investigate the distribution effect on model performance. Meanwhile, we implement Wald/InvGauss and Expon, which provide the best approximate distribution on IMDb datasets.

C. MODEL PERFORMANCE

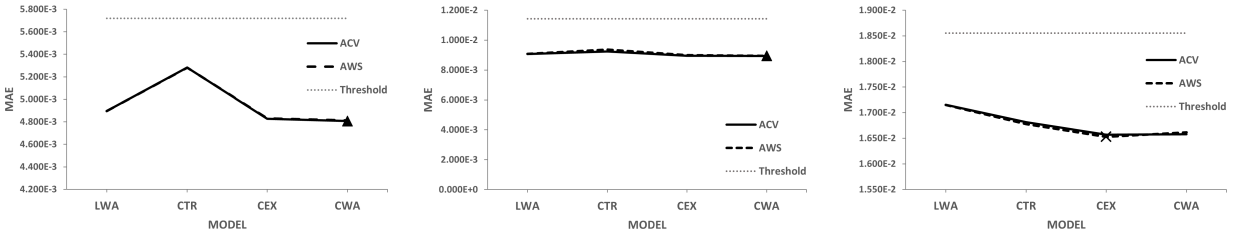
Here, we show that implementing a model adapting to unsuitable distribution tends to give an unacceptable and suboptimal result. We first develop the model that adapts to the best approximate distribution, as in Table 6. We then compare the results of the model adapting to the best approximate distribution along with those of the other baseline models with the average MAE of CTR-ReLU, the acceptance threshold, as mentioned in Subsection V-A. Subsequently, we check the impact of sampling methods.

Table 7 shows that the models adapted to the best approximate distribution always give an acceptable result on Amazon datasets, primarily when implemented in LM (LWA) and CNN (CWA). All miscellaneous and LM-based models give acceptable results when evaluated with AWS on the AD1 dataset, where the average MAE is under the threshold, the average MAE of CTR-ReLU. Meanwhile, with ACV, we find that only SGD, RFR, GBR, ETR, and LWA models provide acceptable results. Models built in XGB and CNN, except CSE, also provide acceptable results on AD1. However, we get fewer models (LWA, CTR, CEX, and CWA) which give acceptable results on AD2 and AD3 datasets.

TABLE 8. Performance of models on IMDb datasets with the ACV and AWS sampling methods. The performance is measured by the average of MAE followed by the standard deviation in the parentheses. We use the result of CTR-ReLU as the acceptance threshold of models.

Model	IMDb Dataset					
	ID1		ID2		ID3	
	ACV	AWS	ACV	AWS	ACV	AWS
Threshold	5.718E-3 (1.17E-3)	5.718E-3 (1.17E-3)	1.142E-2 (4.68E-3)	1.142E-2 (4.68E-3)	1.855E-2 (1.87E-2)	1.855E-2 (1.87E-2)
Misc.						
SGD	8.315E-3 (1.71E-3)	8.353E-3 (1.81E-3)	1.414E-2 (4.08E-3)	1.404E-2 (4.07E-3)	2.240E-2 (1.50E-2)	2.241E-2 (1.51E-2)
OMP	2.342E-2 (7.84E-3)	2.364E-2 (7.93E-3)	4.629E-2 (1.40E-2)	4.654E-2 (1.38E-2)	5.425E-2 (2.61E-2)	5.548E-2 (2.62E-2)
NET	8.415E-3 (1.76E-3)	8.392E-3 (1.81E-3)	1.489E-2 (5.03E-3)	1.492E-2 (5.01E-3)	2.311E-2 (1.48E-2)	2.313E-2 (1.48E-2)
LAS	8.415E-3 (1.76E-3)	8.392E-3 (1.81E-3)	1.489E-2 (5.03E-3)	1.492E-2 (5.01E-3)	2.311E-2 (1.48E-2)	2.313E-2 (1.48E-2)
LAR	4.95E+9 (9.29E+9)	4.95E+9 (9.29E+9)	7.19E+1 (2.00E+13)	1.01E+11 (2.83E+11)	5.00E+97 (1.41E+98)	5.00E+97 (1.41E+98)
LAL	8.415E-3 (1.76E-3)	8.392E-3 (1.81E-3)	1.489E-2 (5.03E-3)	1.492E-2 (5.01E-3)	2.311E-2 (1.48E-2)	2.313E-2 (1.48E-2)
ARD	2.611E-2 (1.08E-2)	2.627E-2 (1.08E-2)	4.062E-2 (6.15E-3)	4.091E-2 (5.66E-3)	4.560E-2 (2.39E-2)	4.516E-2 (2.30E-2)
RFR	9.821E-3 (2.04E-3)	9.619E-3 (2.12E-3)	1.703E-2 (5.43E-3)	1.659E-2 (5.39E-3)	2.545E-2 (1.47E-2)	2.534E-2 (1.49E-2)
GBR	8.710E-3 (1.60E-3)	8.877E-3 (2.03E-3)	1.834E-2 (8.91E-3)	1.801E-2 (8.25E-3)	2.844E-2 (1.81E-2)	2.795E-2 (1.71E-2)
ETR	7.870E-3 (1.71E-3)	7.844E-3 (1.74E-3)	1.650E-2 (5.16E-3)	1.635E-2 (5.36E-3)	2.327E-2 (1.57E-2)	2.369E-2 (1.56E-2)
LM						
LSE	8.430E-3 (1.76E-3)	8.405E-3 (1.80E-3)	1.492E-2 (5.05E-3)	1.494E-2 (5.03E-3)	2.312E-2 (1.48E-2)	2.313E-2 (1.48E-2)
LTR	8.430E-3 (1.76E-3)	8.405E-3 (1.80E-3)	1.492E-2 (5.05E-3)	1.494E-2 (5.03E-3)	2.312E-2 (1.48E-2)	2.313E-2 (1.48E-2)
LEX	8.173E-3 (1.57E-3)	8.152E-3 (1.61E-3)	1.439E-2 (4.56E-3)	1.441E-2 (4.54E-3)	2.281E-2 (1.48E-2)	2.282E-2 (1.48E-2)
LWA	**4.897E-3** (1.17E-3)	**4.896E-3** (1.17E-3)	**9.078E-3** (4.45E-3)	**9.084E-3** (4.44E-3)	**1.715E-2** (1.80E-2)	**1.715E-2** (1.80E-2)
XGB						
XSE	1.079E-2 (2.11E-3)	1.084E-2 (2.25E-3)	2.109E-2 (6.82E-3)	2.063E-2 (7.16E-3)	2.842E-2 (1.66E-2)	2.885E-2 (1.59E-2)
XTR	1.026E-2 (1.93E-3)	1.031E-2 (2.13E-3)	2.047E-2 (6.67E-3)	2.004E-2 (6.99E-3)	2.752E-2 (1.62E-2)	2.811E-2 (1.54E-2)
XEX	1.014E-2 (2.71E-3)	1.024E-2 (2.68E-3)	1.778E-2 (4.93E-3)	1.735E-2 (5.35E-3)	2.561E-2 (1.59E-2)	2.600E-2 (1.59E-2)
XWA	8.357E-3 (1.95E-3)	8.308E-3 (1.96E-3)	1.593E-2 (5.11E-3)	1.614E-2 (5.00E-3)	2.444E-2 (1.43E-2)	2.455E-2 (1.42E-2)
CNN						
CSE	1.375E-2 (3.29E-3)	1.368E-2 (3.98E-3)	3.569E-2 (3.26E-2)	3.556E-2 (3.27E-2)	1.456E-1 (1.77E-1)	1.033E-1 (1.46E-1)
CTR	**5.281E-3** (1.09E-3)	**5.281E-3** (1.09E-3)	**9.234E-3** (4.64E-3)	**9.372E-3** (4.59E-3)	**1.681E-2** (1.82E-2)	**1.678E-2** (1.82E-2)
CEX	**4.828E-3** (1.15E-3)	**4.830E-3** (1.16E-3)	**8.957E-3** (4.62E-3)^c	**8.996E-3** (4.60E-3)	**1.657E-2** (1.81E-2)	**1.653E-2** (1.80E-2)^b
CWA	**4.808E-3** (1.14E-3)^a	**4.813E-3** (1.14E-3)^b	**8.950E-3** (4.54E-3)^a	**8.954E-3** (4.55E-3)^b	**1.658E-2** (1.82E-2)	**1.662E-2** (1.82E-2)

Note: *The boldface means the model has an acceptable result. Results with* ^{a} *are significant for* $p \leq 0.01$, ^{b} *for* $p \leq 0.05$ *and* ^{c} *for* $p \leq 0.1$.



(a) ID1 (b) ID2 (c) ID3

FIGURE 5. Performance of models under the threshold line on IMDb dataset. Distribution-adapted models in LM and CNN do not always cross the line.

Overall, CWA gives the best results on Amazon datasets, as shown in Table 7 and Fig. 4. These results follow the pattern of those of distribution identification with all approaches, as shown in Table 6. MSE, AIC, and KS approaches give the same best approximate distribution, Wald on Amazon datasets.

We also find that LWA, CTR, CEX, and CWA models give acceptable results on IMDb datasets as shown in Table 8 and Fig. 5. On ID1 and ID2, CWA gives the best result when evaluated with ACV. On the smallest dataset, ID3, CEX has the best achievement with AWS. These results are in the same pattern as the best approximate distribution of the MSE score, as shown in Table 6.

The best model on Amazon datasets CWA is achieved when evaluated with AWS. In addition, on two most extensive datasets: AD1 and AD2, model evaluation with AWS has a significant impact, as shown in Figs. 4(a) and 4(b). However, on the AD3 and IMDb datasets, model evaluation with ACV has no significant difference compared to AWS, as shown in Figs. 4(c) and 5.

Furthermore, we analyze effects of distributions to which the model is adapted, the sampling methods used, and the time consumed by the model.

D. EFFECT OF DISTRIBUTION

We show the effect of distribution on the model performance in Figs. 6 to 8 and answer Q2 in Section I. We also show the improvement in the model performance when models follow the identified approximate distribution in Table 9.

The models that adapted to the best approximate distribution LWA give acceptable results on all datasets. In addition, the performance of the model built in LM follows the rank of the identified distributions, as shown in Table 9. The results on Amazon datasets shown in Figs. 6(a) to 6(c) follow the

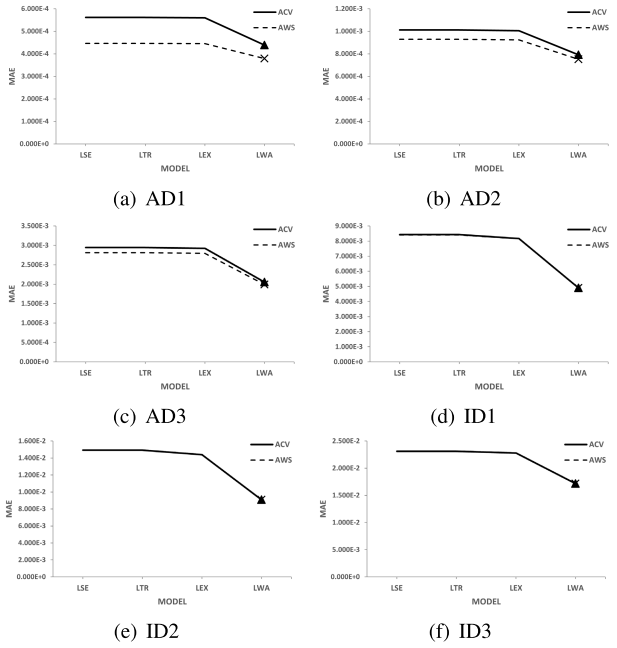


FIGURE 6. Performance of models implemented in LM. The version follows the best approximate distribution with the KS score.

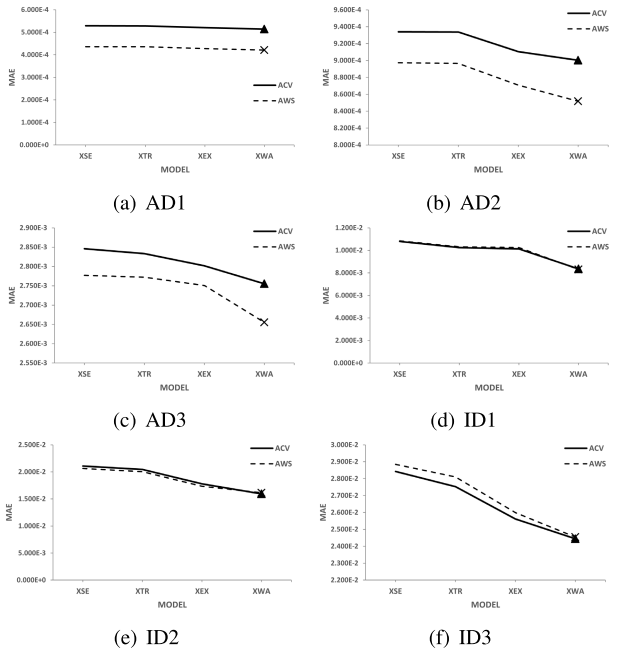


FIGURE 7. Performance of models implemented in XGB. The version follows the best approximate distribution with the KS score.

MSE, AIC and KS scores as in Table 6. On the other hand, the results on IMDb datasets, as shown in Figs. 6(d) to 6(f), follow KS on all datasets, MSE (ID1 and ID2), and AIC (ID2).

The effect of identified distribution, which was used in XGB, is perfectly shown in Fig. 7. However, distribution-adapted models have only a slight effect on AD1. Still, overall, XEX and XWA models gave a consistent effect according to the identification rank of the distributions,

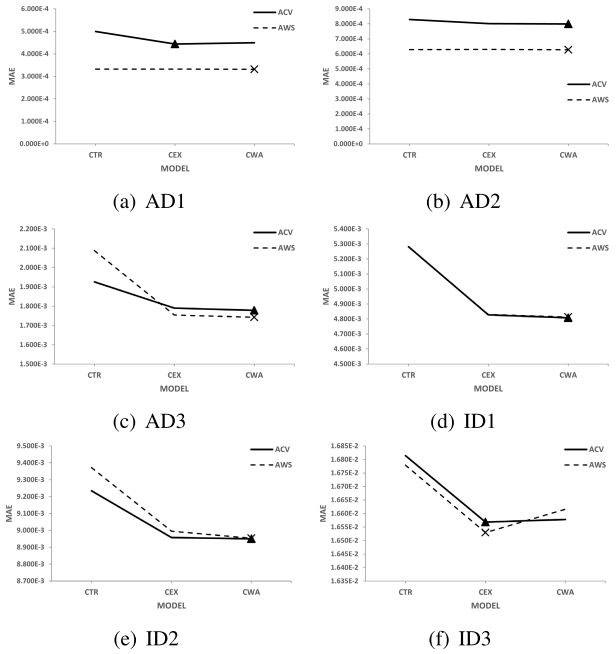


FIGURE 8. Performance of models implemented in CNN. The version follows the best approximate distribution with the MSE score.

TABLE 9. Performance improvement as effect of distribution-adapted model compared to baseline models.

Dataset	Sampling Methods	Model					
		LEX	LWA	XEX	XWA	CEX	CWA
AD1	ACV	0.3%	**21.8%**	1.4%	2.7%	11.2%	10.1%
	AWS	0.2%	**15.1%**	1.8%	3.3%	(0.1%)	0.2%
AD2	ACV	0.6%	**21.9%**	2.5%	3.6%	3.5%	3.6%
	AWS	0.5%	**19.2%**	2.9%	5.0%	(0.4%)	0.2%
AD3	ACV	0.7%	**30.3%**	1.1%	2.7%	7.1%	7.7%
	AWS	0.8%	**29.3%**	0.8%	4.2%	16.0%	16.5%
ID1	ACV	3.0%	**41.9%**	1.2%	18.5%	8.6%	9.0%
	AWS	3.0%	**41.7%**	0.7%	19.4%	8.5%	8.9%
ID2	ACV	3.6%	**39.1%**	13.1%	22.2%	3.0%	3.1%
	AWS	3.6%	**39.2%**	13.4%	19.5%	4.0%	4.5%
ID3	ACV	1.3%	**25.8%**	6.9%	11.2%	1.5%	1.4%
	AWS	1.3%	**25.8%**	7.5%	12.7%	1.5%	1.0%

Note: *The result in parentheses means the model gives negative improvement.*

as shown in Table 6, especially with the KS score. Those results also follow the rank by MSE (despite ID3) and AIC (despite ID1 and ID3). Table 9 confirms those results.

While the model performance in LM and XGB fully follows the rank by the KS score, the model performance in CNN entirely follows the rank by MSE. Tables 7 and 8 show that the distribution-adapted models give a significant drop on MAE compared to CSE. Fig. 8 shows that CEX and CWA also perform better than CTR, as in Table 9, except on AD1 and AD2 when evaluated in AWS. On AD1 and AD2, CTR performs better than CEX when evaluated with AWS. Following the distribution identification with the MSE scores, as in Table 6, CEX performs the best on ID3, while CWA on the rest.

The implementation of distribution-adapted models affects LM more than other machine learning, in Table 9. LWA gives more than 15% improvement compared to the best model

TABLE 10. Performance improvement as effect of evaluation with AWS sampling methods compared to ACV sampling methods.

Model	Dataset					
	AD1	AD2	AD3	ID1	ID2	ID3
LM						
LSE	20.5%	8.1%	4.4%	0.3%	(0.2%)	(0.0%)
LTR	20.5%	8.1%	4.4%	0.3%	(0.2%)	(0.0%)
LEX	20.4%	8.0%	**4.5%**	0.3%	(0.2%)	(0.0%)
LWA	13.7%	5.0%	3.0%	0.0%	(0.1%)	0.0%
XGB						
XSE	17.6%	3.9%	2.4%	(0.5%)	2.2%	(1.5%)
XTR	17.5%	4.0%	2.2%	(0.6%)	2.1%	(2.1%)
XEX	17.8%	4.4%	1.8%	(1.0%)	**2.4%**	(1.5%)
XWA	18.0%	5.4%	3.6%	**0.6%**	(1.3%)	(0.4%)
CNN						
CSE	(237.1%)	(39.7%)	(20.4%)	**0.6%**	0.4%	**29.1%**
CTR	**33.5%**	**24.3%**	(8.4%)	0.0%	(1.5%)	0.2%
CEX	25.1%	21.3%	2.0%	(0.0%)	(0.4%)	0.2%
CWA	26.2%	21.6%	2.0%	(0.1%)	(0.0%)	(0.2%)

adapting a normal distribution. The same pattern with smaller improvement appears in XGB, as in Table 9. Implementation with AWS in ID1 gets the greatest effect. Meanwhile, the implementation with ACV on AD2 gets the least impact. The smaller effect is in CNN, with less than 1% improvement.

Overall, the implementation of the model that adapts to the distribution gives a positive improvement in all machine learning.

E. EFFECT OF SAMPLING METHODS

Next, we answer Q3 in Section I. We calculate the improvement of each model performance with the AWS sampling method compared to ACV, as shown in Table 10.

Table 10 shows that evaluation with AWS affects the model performance on large-size datasets, Amazon. AWS gives significant positive results for all models except for CSE on AD1 to AD3. ACV and AWS have dynamic results on IMDb datasets with no significant difference. The best model on ID1 and ID2 is CWA, achieved with ACV, and on ID3 is CEX when evaluated with AWS.

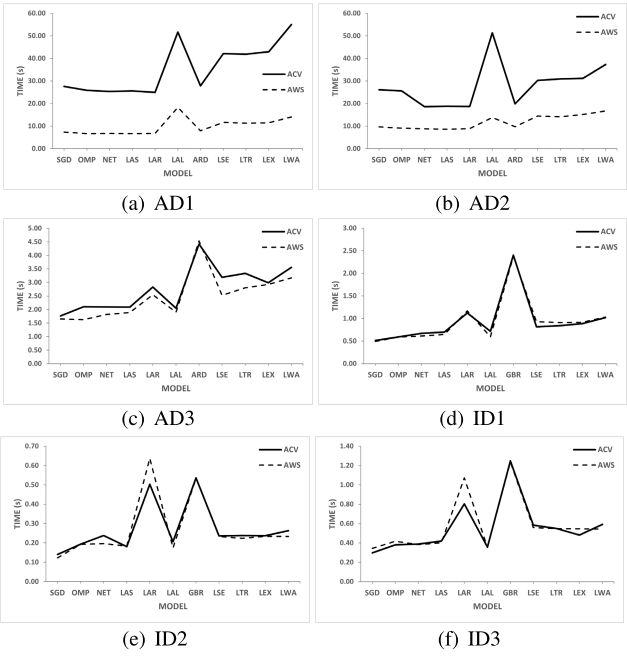
Based on the above results, we can answer Q3 that the model evaluation with AWS improves the performance, especially on large datasets. Moreover, it gives an almost constant improvement in LM and XGB models.

F. TIME CONSUMPTION

Here, we answer Q3 in Section I by comparing the time consumption of each model in each machine learning and dataset. We calculate the time consumed from the start of training the model to obtaining the test results.

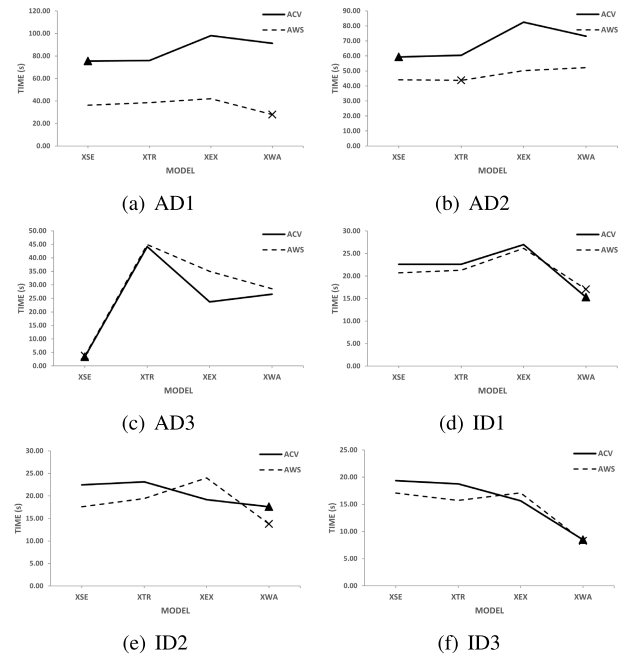
We find that all models run far faster when evaluated in the AWS sampling methods on the two most extensive datasets, AD1 and AD2. We find that distribution-adapted models spend various run times depending on the machine learning, the dataset characteristics, and sampling methods. The details are shown in Appendix C.

In LM, the best approximate distribution model LWA double the time of the fastest model to get a result, as shown in Fig. 9. LAR is the quickest model on the two largest datasets,



(a) AD1 (b) AD2 (c) AD3 (d) ID1 (e) ID2 (f) ID3

FIGURE 9. Model run time in LM.



(a) AD1 (b) AD2 (c) AD3 (d) ID1 (e) ID2 (f) ID3

FIGURE 10. Model run time in XGB.

AD1 and AD2, while SGD is the fastest on the rest. We also find that the model evaluation with AWS reduces the run time by more than 60% compared to ACV on AD1 and AD2. On the rest, using AWS has no significant effect on the time consumed to run the model.

In XGB, the best approximate distribution-adapted model, XWA, spends less than other models on IMDb datasets, as shown in Figs. 10(d) to 10(f). On Amazon datasets, XWA spends more time than XSE, as shown in

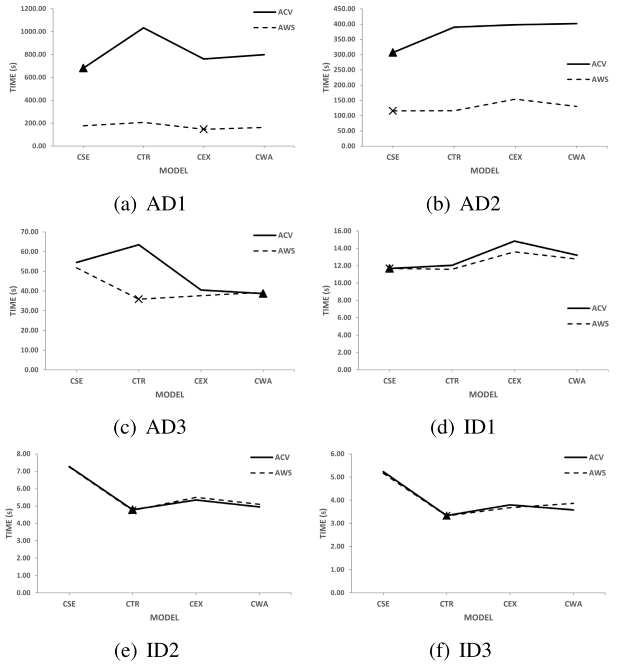


FIGURE 11. Model run time in CNN.

Figs. 10(b) and 10(c), except on AD1 with AWS. Moreover, using AWS reduces the time consumption up to 75% on AD1 and 30% on AD2.

CNN is the most time-consuming machine learning, as shown in Figs. 11(a) to 11(c), which is almost 10 times XGB and 20 times LM on Amazon datasets. However, on IMDb datasets, CNN models perform faster than XGB, as shown in Figs. 11(d) to 11(f). Among CNN models, CWA and CEX reach a level with the fastest models on all datasets, despite AD2 and ID1, as shown in Figs. 11(a), 11(c), 11(e) and 11(f). On AD2, CWA and CEX increase slightly compared to CSE, but gain a level with CTR, as shown in Fig. 11(b). Meanwhile, Fig. 11(d) shows that CEX reaches the top when CWA consumes slightly more time than CSE and CTR. Consistent with the LM and XGB, AWS reduces the time consumption by more than 60% on AD1 and AD2 for all models.

G. DISCUSSION

Previous studies commonly use the helpfulness rating as a dependent variable since their datasets, such as Amazon.com 2014 [37], have helpful and total votes to measure helpfulness. Moreover, their focus is on model factors' contribution to helpfulness, and many factors appear as independent variables in Table 1. So, we cannot make a direct comparison with previous research.

In this research, we use Amazon.com 2018 [22] as an updated version of Amazon.com 2014 [37], which has dropped total votes. With the result, we use the helpful votes as a dependent variable, even on IMDb dataset [23]. To make a comparison with the state-of-the-art helpfulness rating prediction models [9], [30], we rerun them in the helpful votes on Amazon.com [22], and IMDb [23] dataset. Our proposed framework, especially with LM and CNN approaches, does not have poor performance compared to previous studies, as shown in Table 11.

TABLE 11. Comparison with previous research.

Reference	Machine Learning	Dataset		MAE	MSE
Malik & Hussain [30]	Ensemble	Amazon.com [37]		9.910E-2[a]	1.990E-2[a]
		Amazon.com [22]	AD1	4.113E-4	
			AD2	8.845E-4	
			AD3	2.664E-3	
		IMDb [23]	ID1	8.405E-3	
			ID2	1.492E-2	
			ID3	2.312E-2	
Saumya et al. [9][a]	CNN	Amazon.com [37]			2.130E-1[a]
		Amazon.com [22]	AD1	1.045E-3	
			AD2	2.480E-3	
			AD3	1.121E-2	
		IMDb [23]	ID1	1.368E-2	
			ID2	3.556E-2	
			ID3	1.033E-1	
Proposed	LM	Amazon.com [22]	AD1	**3.778E-4**	
			AD2	**7.510E-4**	
			AD3	**1.991E-3**	
		IMDb [23]	ID1	**4.896E-3**	
			ID2	**9.078E-3**	
			ID3	**1.715E-2**	
	XGB	Amazon.com [22]	AD1	**4.215E-4**	
			AD2	**8.518E-4**	
			AD3	**2.665E-3**	
		IMDb [23]	ID1	**8.308E-3**	
			ID2	**1.593E-2**	
			ID3	**2.444E-2**	
	CNN	Amazon.com [22]	AD1	**3.315E-4**	
			AD2	**6.273E-4**	
			AD3	**1.743E-3**	
		IMDb [23]	ID1	**4.808E-2**	
			ID2	**8.950E-2**	
			ID3	**1.653E-2**	

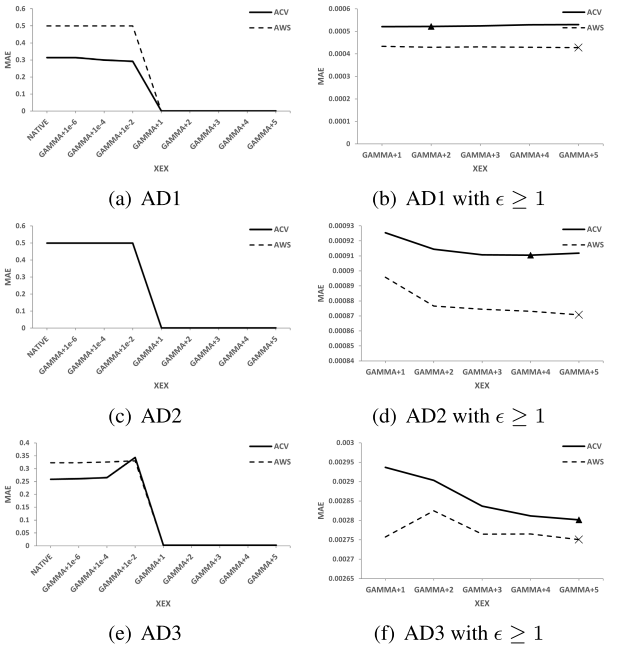
Note: [a] *The result on the helpfulness rating prediction.*

VI. CONCLUSION AND FUTURE WORK

This study discussed the benefits of checking the distribution of helpful votes of reviews in a dataset. It was proved that the distribution of helpful votes significantly affects the model performance. The performances consistently follow the rank of distribution identification results, especially when implementing LM and XGB.

The experimental results illustrated that the helpful votes are not statistically distributed in a continuous distribution. Meanwhile, MSE and AIC consistently show that Wald is the best approximate distribution of the helpful votes on Amazon datasets. This result follows the calibrator of the KS score. On the other hand, the best approximate distribution is dynamic among Expon, InvGauss, and Wald on IMDb datasets. MSE and AIC have a distinct result on ID1, where MSE gives Wald while AIC gives Expon. Both approaches give Wald and Expon for ID2 and ID3, respectively. These results do not follow KS, which has InvGauss for ID1 and ID2, and Wald for ID3.

Models adapting to Wald distribution are significantly improved compared to the other models, following the best approximate distribution on Amazon datasets. On IMDb datasets, Wald distribution-adapted model, CWA, gives the

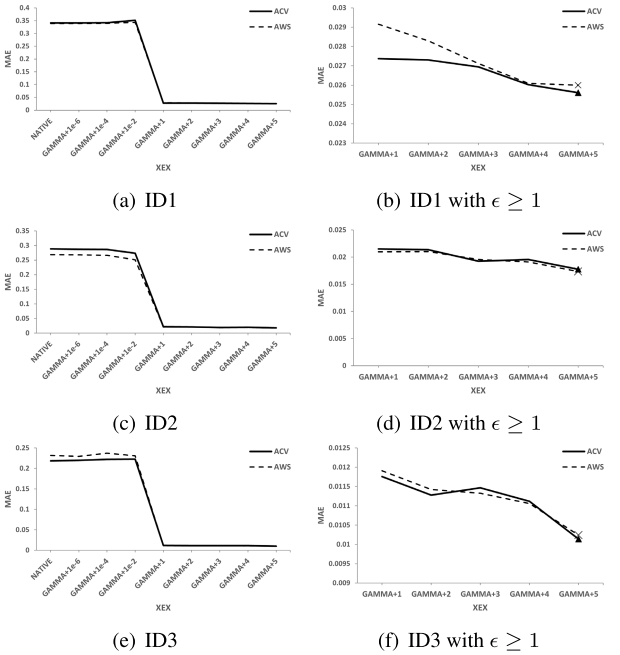


(a) AD1 (b) AD1 with $\epsilon \geq 1$

(c) AD2 (d) AD2 with $\epsilon \geq 1$

(e) AD3 (f) AD3 with $\epsilon \geq 1$

FIGURE 12. XEX with GAMMA+ϵ performance compared to the native objective function of Gamma/Expon model on Amazon datasets. GAMMA+ϵ can handle normalized helpful votes better when $\epsilon \geq 1$.

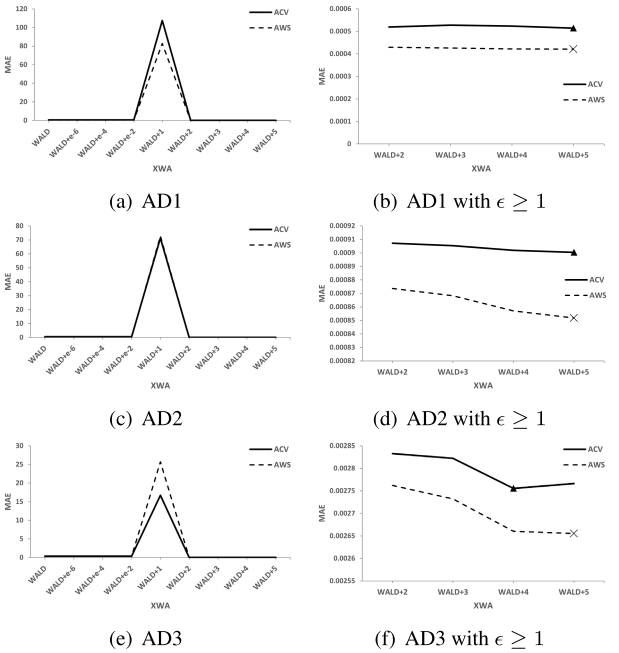


(a) ID1 (b) ID1 with $\epsilon \geq 1$

(c) ID2 (d) ID2 with $\epsilon \geq 1$

(e) ID3 (f) ID3 with $\epsilon \geq 1$

FIGURE 13. XEX with GAMMA+ϵ performance compared to the native objective function of Gamma/Expon model on IMDb datasets. GAMMA+ϵ can handle normalized helpful votes better when $\epsilon \geq 1$.

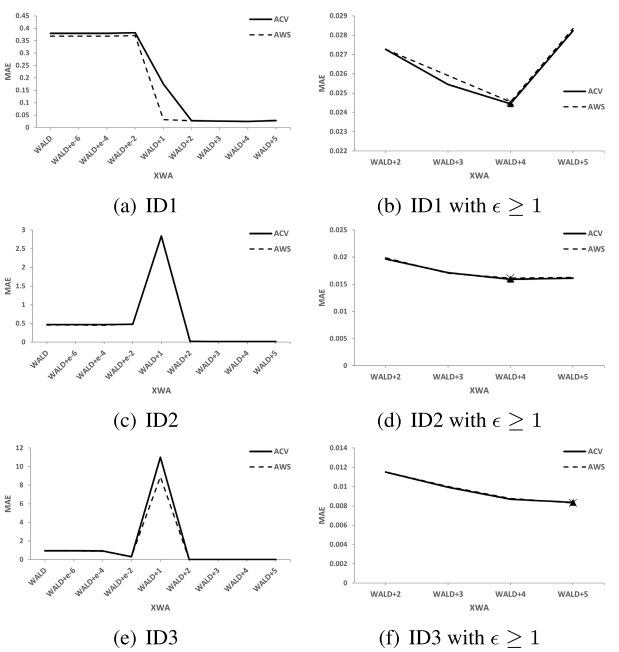


(a) AD1 (b) AD1 with $\epsilon \geq 1$

(c) AD2 (d) AD2 with $\epsilon \geq 1$

(e) AD3 (f) AD3 with $\epsilon \geq 1$

FIGURE 14. XWA with WALD+ϵ performance compared with WALD on Amazon datasets. WALD+ϵ can handle normalized helpful votes better when $\epsilon \geq 1$.



(a) ID1 (b) ID1 with $\epsilon \geq 1$

(c) ID2 (d) ID2 with $\epsilon \geq 1$

(e) ID3 (f) ID3 with $\epsilon \geq 1$

FIGURE 15. XWA with WALD+ϵ performance compared with WALD on IMDb datasets. WALD+ϵ can handle normalized helpful votes better when $\epsilon \geq 1$.

best result on ID1 and ID2, while Expon distribution-adapted model, CEX, is on ID3. Those results are the same pattern as the distribution identified by the MSE score.

When predicting the number of helpful votes, it is important to take into account the sampling time elapsed since the review was posted. It was proved that the model evaluation with AWS, an adjusted window size in the TBS method, has a greater effect with much less training data than using ACV, especially on large and medium size datasets. Moreover, AWS provides slightly better results on

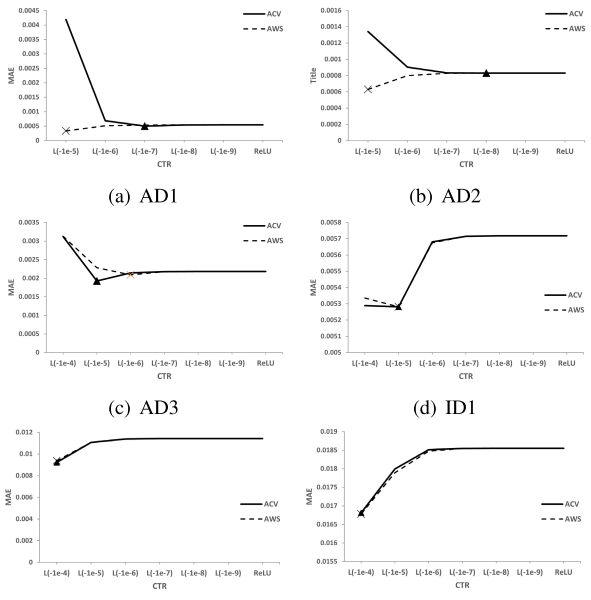
TABLE 12. Average model run time spent on each iteration on Amazon datasets with the standard deviation in parentheses.

Model	Time spends on Amazon datasets (second)					
	AD1		AD2		AD3	
	ACV	AWS	ACV	AWS	ACV	AWS
Misc.						
SGD	27.55 (14.62)	7.32 (0.58)	26.11 (10.36)	9.67 (1.30)	1.77 (0.78)	1.66 (0.73)
OMP	25.88 (11.38)	6.67 (1.03)	25.61 (12.95)	9.06 (1.11)	2.10 (0.80)	1.63 (0.69)
NET	25.34 (10.53)	6.71 (1.12)	18.65 (7.39)	8.81 (1.51)	2.09 (1.12)	1.82 (0.88)
LAS	25.60 (10.92)	6.66 (1.02)	18.78 (7.65)	8.58 (1.41)	2.09 (0.85)	1.89 (0.83)
LAR	24.91 (10.66)	6.74 (0.99)	18.73 (7.37)	8.90 (1.20)	2.83 (0.90)	2.55 (0.74)
LAL	51.68 (17.47)	18.30 (2.59)	51.38 (24.83)	13.78 (3.01)	2.05 (1.01)	1.90 (0.95)
ARD	27.85 (11.68)	7.90 (1.10)	19.85 (8.17)	9.74 (1.51)	4.42 (3.20)	4.54 (3.72)
RFR	7823.93 (4831.37)	1123.74 (246.00)	3715.10 (2183.66)	1343.77 (461.87)	219.45 (133.52)	210.68 (119.13)
GBR	403.86 (224.11)	63.67 (11.97)	212.21 (115.04)	82.44 (23.38)	12.24 (7.87)	10.58 (6.23)
ETR	7762.40 (4510.04)	1179.52 (226.69)	3518.20 (2146.40)	1277.75 (452.16)	241.51 (142.26)	219.64 (121.31)
LM						
LSE	42.09 (17.33)	11.66 (1.70)	30.24 (11.82)	14.46 (2.38)	3.19 (1.15)	2.52 (1.15)
LTR	41.91 (17.19)	11.34 (1.42)	30.89 (12.23)	14.11 (2.18)	3.33 (1.46)	2.81 (1.19)
LEX	42.92 (17.70)	11.46 (1.82)	31.20 (12.40)	15.10 (2.12)	2.99 (1.41)	2.93 (1.24)
LWA	54.97 (22.56)	14.07 (2.53)	37.30 (15.27)	16.72 (2.73)	3.56 (1.29)	3.17 (1.51)
XGB						
XSE	75.56 (17.87)	36.15 (2.17)	59.25 (13.00)	44.03 (3.29)	3.36 (1.05)	3.75 (0.93)
XTR	76.06 (19.53)	38.62 (2.71)	60.44 (12.17)	43.68 (2.62)	44.15 (3.43)	44.92 (1.97)
XEX	98.10 (21.00)	42.09 (4.07)	82.50 (16.25)	50.15 (3.19)	23.69 (4.14)	35.03 (4.00)
XWA	91.34 (26.23)	27.83 (5.96)	73.11 (16.67)	52.13 (2.50)	26.58 (3.38)	28.59 (6.36)
CNN						
CSE	681.86 (405.08)	177.65 (28.68)	306.96 (169.17)	115.15 (24.88)	54.51 (27.31)	51.76 (23.57)
CTR	1033.62 (619.86)	206.18 (26.37)	389.91 (214.66)	115.98 (28.91)	63.49 (30.92)	35.87 (15.92)
CEX	761.02 (446.65)	146.19 (32.61)	398.11 (219.32)	153.83 (37.77)	40.53 (19.27)	37.64 (17.85)
CWA	798.45 (470.65)	163.31 (26.93)	401.52 (226.27)	130.31 (32.15)	38.72 (17.74)	39.36 (17.21)

small-size datasets. In addition, AWS also positively affects model performance on average when implementing LM, XGB, and CNN.

Evaluation with the AWS sampling method on two large datasets, AD1 and AD2, decreases the time consumption significantly for all models. On the other hand, the time consumed by a model that follows the best approximate distribution to produce results is variable. It depends on machine learning, characteristics of the data set, and sampling methods. In CNN, normal distribution-adapted models perform faster: CSE on two large datasets, AD1 and AD2, and CTR on two small datasets, ID2 and ID3. However, models adapting to Expon and Wald distributions spend not far different from the fastest model. In XGB, Wald distribution-adapted models spend not far different on Amazon datasets and even faster than models adapting to other distributions on IMDb datasets. On the other hand, LWA consumes double compared to the quickest model in LM and miscellaneous models. However, it is still under a minute on the largest dataset and even a second on IMDb datasets.

The best approximate distribution is identified by measuring the distribution of whole helpful vote reviews in each dataset. Adaptively changing the distribution identification on the training set will be challenging. Moreover, there is a minor difference in the order of AIC, MSE, and KS scores on InvGauss and Gamma. Investigating the effect of the metric on the other datasets also becomes a further task. Considering the advantage of the RNN-based model in sequential data, developing an RNN-based model for helpful votes prediction also becomes a challenge.



(a) AD1 (b) AD2 (c) AD3 (d) ID1 (e) ID2 (f) ID3

FIGURE 16. CTR is sensitive to the change of the LeakyReLU coefficient. CTR with AWS has the same coefficient of the LeakyReLU as with ACV on IMDb datasets.

APPENDIX A GAMMA AND WALD DEVIANCE TRANSLATION EFFECT ON XGB

See Figures 12–15.

APPENDIX B LeakyReLU COEFFICIENT FOR CTR

See Figure 16.

TABLE 13. Average model run time spent on each iteration on IMDb datasets with the standard deviation in parentheses.

Model	Time spends on IMDb datasets (second)					
	ID1		ID2		ID3	
	ACV	AWS	ACV	AWS	ACV	AWS
Misc.						
SGD	0.51 (0.23)	0.49 (0.19)	0.30 (0.12)	0.34 (0.14)	0.14 (0.05)	0.13 (0.05)
OMP	0.59 (0.25)	0.59 (0.26)	0.38 (0.16)	0.42 (0.12)	0.18 (0.06)	0.18 (0.06)
NET	0.67 (0.32)	0.61 (0.28)	0.39 (0.14)	0.38 (0.15)	0.20 (0.06)	0.20 (0.06)
LAS	0.70 (0.35)	0.64 (0.26)	0.42 (0.19)	0.40 (0.17)	0.18 (0.05)	0.20 (0.05)
LAR	1.12 (0.34)	1.17 (0.43)	0.80 (0.19)	1.07 (0.33)	0.81 (0.37)	0.59 (0.19)
LAL	0.72 (0.35)	0.59 (0.25)	0.36 (0.15)	0.35 (0.13)	0.18 (0.05)	0.19 (0.07)
ARD	27.37 (44.15)	13.79 (10.96)	15.74(10.02)	17.53 (16.00)	3.73 (2.83)	3.90 (2.97)
RFR	47.24 (42.94)	23.35 (16.28)	6.40 (4.45)	6.34 (4.36)	3.54 (2.42)	3.46 (2.38)
GBR	2.40 (1.23)	2.37 (1.20)	1.25 (0.63)	1.23 (0.62)	0.62 (0.31)	0.62 (0.31)
ETR	21.50 (12.95)	21.17 (12.79)	6.82 (4.36)	6.71 (4.27)	5.26 (3.70)	5.16 (3.58)
LM						
LSE	0.81 (0.39)	0.93 (0.41)	0.58 (0.27)	0.56 (0.24)	0.24 (0.08)	0.23 (0.09)
LTR	0.84 (0.34)	0.91 (0.40)	0.55 (0.24)	0.55 (0.24)	0.23 (0.08)	0.20 (0.08)
LEX	0.89 (0.43)	0.91 (0.41)	0.48 (0.17)	0.54 (0.24)	0.25 (0.09)	0.25 (0.10)
LWA	1.02 (0.41)	1.03 (0.43)	0.59 (0.24)	0.54 (0.28)	0.24 (0.09)	0.16 (0.04)
XGB						
XSE	22.62 (3.55)	20.70 (5.23)	22.47 (3.37)	17.58 (4.38)	19.37 (3.98)	17.07 (5.40)
XTR	22.61 (4.86)	21.28 (5.90)	23.13 (3.25)	19.40 (4.57)	18.78 (4.95)	15.71 (4.11)
XEX	26.97 (4.35)	26.14 (4.90)	19.19 (6.21)	23.99 (6.72)	15.63 (6.09)	17.09 (8.21)
XWA	15.30 (5.01)	17.04 (6.64)	17.58 (5.73)	13.79 (4.79)	8.49 (3.12)	8.28 (2.99)
CNN						
CSE	11.67 (4.07)	11.66 (4.05)	7.27 (2.16)	7.24 (2.08)	5.23 (0.89)	5.17 (1.13)
CTR	12.05 (4.27)	11.58 (4.30)	4.78 (1.34)	4.74 (1.58)	3.34 (0.88)	3.32 (0.69)
CEX	14.84 (5.33)	13.59 (4.03)	5.35 (1.69)	5.50 (1.74)	3.80 (0.95)	3.68 (0.75)
CWA	13.21 (4.42)	12.75 (4.30)	4.95 (1.53)	5.10 (1.66)	3.58 (0.73)	3.87 (0.79)

APPENDIX C THE TIME CONSUMPTION

See Tables 12–13.

ACKNOWLEDGMENT

The authors would like to thank Seita Shimada and Yoshiaki Matsukawa, Rakuten Card Co. Ltd., for their helpful comments on this research.

REFERENCES

[1] Y. Zhou and S. Yang, "Roles of review numerical and textual characteristics on review helpfulness across three different types of reviews," *IEEE Access*, vol. 7, pp. 27769–27780, 2019.

[2] Y. Heng, Z. Gao, Y. Jiang, and X. Chen, "Exploring hidden factors behind online food shopping from Amazon reviews: A topic mining approach," *J. Retailing Consum. Services*, vol. 42, pp. 161–168, May 2018.

[3] X. Sun, M. Han, and J. Feng, "Helpfulness of online reviews: Examining review informativeness and classification thresholds by search products and experience products," *Decis. Support Syst.*, vol. 124, Sep. 2019, Art. no. 113099.

[4] J. E. Fresneda and D. Gefen, "A semantic measure of online review helpfulness and the importance of message entropy," *Decis. Support Syst.*, vol. 125, Oct. 2019, Art. no. 113117.

[5] Y.-C. Chou, H. H.-C. Chuang, and T.-P. Liang, "Elaboration likelihood model, endogenous quality indicators, and online review helpfulness," *Decis. Support Syst.*, vol. 153, Feb. 2022, Art. no. 113683.

[6] Z. Liu and S. Park, "What makes a useful online review? Implication for travel product websites," *Tourism Manage.*, vol. 47, pp. 140–151, Apr. 2015.

[7] S. Lu, J. Wu, and S.-L. Tseng, "How online reviews become helpful: A dynamic perspective," *J. Interact. Marketing*, vol. 44, pp. 17–28, Nov. 2018.

[8] S. M. Mudambi and D. Schuff, "Research note: What makes a helpful online review? A study of customer reviews on Amazon.Com," *MIS Quart.*, vol. 34, no. 1, p. 185, 2010.

[9] S. Saumya, J. P. Singh, and Y. K. Dwivedi, "Predicting the helpfulness score of online reviews using convolutional neural network," *Soft Comput.*, vol. 24, no. 15, pp. 10989–11005, Feb. 2019.

[10] Q. V. Le, A. J. Smola, T. Gärtner, and Y. Altun, "Transductive Gaussian process regression with automatic model selection," in *Proc. 17th Eur. Conf. Mach. Learn.* Berlin, Germany: Springer-Verlag, 2006, pp. 306–317, doi: 10.1007/11871842_31.

[11] P. K. Dunn and G. K. Smyth, *Generalized Linear Models With Examples in R* (Springer Texts in Statistics). New York, NY, USA: Springer, 2018.

[12] P. X.-K. Song, *Correlated Data Analysis: Modeling, Analytics, and Applications* (Springer Series in Statistics). New York, NY, USA: Springer-Verlag, 2007.

[13] J. Ding, V. Tarokh, and Y. Yang, "Model selection techniques: An overview," *IEEE Signal Process. Mag.*, vol. 35, no. 6, pp. 16–34, Nov. 2018.

[14] H. Pham, "A new criterion for model selection," *Mathematics*, vol. 7, no. 12, p. 1215, Dec. 2019.

[15] D. W. Scott, "On optimal and data-based histograms," *Biometrika*, vol. 66, no. 3, pp. 605–610, Dec. 1979.

[16] N. Smirnov, "Table for estimating the goodness of fit of empirical distributions," *Ann. Math. Statist.*, vol. 19, no. 2, pp. 279–281, Jun. 1948.

[17] D. S. Dimitrova, V. K. Kaishev, and S. Tan, "Computing the Kolmogorov–Smirnov distribution when the underlying CDF is purely discrete, mixed, or continuous," *J. Stat. Softw.*, vol. 95, no. 10, pp. 1–42, 2020.

[18] R. Simard and P. L'Ecuyer, "Computing the two-sided Kolmogorov–Smirnov distribution," *J. Stat. Softw.*, vol. 39, no. 11, pp. 1–18, 2011.

[19] R. Saptono and T. Mine, "Time-based sampling methods for detecting helpful reviews," in *Proc. IEEE/WIC/ACM Int. Joint Conf. Web Intell. Intell. Agent Technol. (WI-IAT)*, Melbourne, VIC, Australia, Dec. 2020, pp. 508–513.

[20] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, San Francisco, CA, USA, Jun. 2016, pp. 785–794.

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

[22] J. Ni, J. Li, and J. McAuley, "Justifying recommendations using distantly-labeled reviews and fine-grained aspects," in *Proc. Conf. Empirical Methods Natural Lang. Process., 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, Hong Kong, 2019, pp. 188–197.

[23] A. Pal, A. Barigidad, and A. Mustafi. (2020). *IMDB Movie Reviews Dataset*. [Online]. Available: https://dx.doi.org/10.21227/zm1y-b270

[24] J. Chen, C. Zhang, and Z. Niu, "Identifying helpful online reviews with word embedding features," in *Knowledge Science, Engineering and Management*, vol. 9983, F. Lehner and N. Fteimi, Eds. Cham, Switzerland: Springer, Oct. 2016, pp. 123–133.
[25] G. Ren and T. Hong, "Examining the relationship between specific negative emotions and the perceived helpfulness of online reviews," *Inf. Process. Manage.*, vol. 56, no. 4, pp. 1425–1438, Jul. 2019.
[26] S.-H. Wu, Y.-H. Hsieh, L.-P. Chen, P.-C. Yang, and L. Fanghuizhu, "Temporal model of the online customer review helpfulness prediction with regression methods," in *Influence Behavior Analysis in Social Networks and Social Media*, M. Kaya and R. Alhajj, Eds. Cham, Switzerland: Springer, 2019, pp. 27–38.
[27] Y. Kang and L. Zhou, "Helpfulness assessment of online reviews: The role of semantic hierarchy of product features," *ACM Trans. Manage. Inf. Syst.*, vol. 10, no. 3, pp. 1–18, Nov. 2019.
[28] F. Wang and S. Karimi, "This product works well (for me): The impact of first-person singular pronouns on online review helpfulness," *J. Bus. Res.*, vol. 104, pp. 283–294, Nov. 2019.
[29] M. Siering, J. Muntermann, and B. Rajagopalan, "Explaining and predicting online review helpfulness: The role of content and reviewer-related signals," *Decis. Support Syst.*, vol. 108, pp. 1–12, Apr. 2018.
[30] M. S. I. Malik and A. Hussain, "Exploring the influential reviewer, review and product determinants for review helpfulness," *Artif. Intell. Rev.*, vol. 53, no. 1, pp. 407–427, Jan. 2020.
[31] J. Yi and Y. K. Oh, "The informational value of multi-attribute online consumer reviews: A text mining approach," *J. Retailing Consum. Services*, vol. 65, Mar. 2022, Art. no. 102519.
[32] M. G. Majumder, S. D. Gupta, and J. Paul, "Perceived usefulness of online customer reviews: A review mining approach using machine learning & exploratory data analysis," *J. Bus. Res.*, vol. 150, pp. 147–164, Nov. 2022.
[33] E. Bigne, C. Ruiz, A. Cuenca, C. Perez, and A. Garcia, "What drives the helpfulness of online reviews? A deep learning study of sentiment analysis, pictorial content and reviewer expertise for mature destinations," *J. Destination Marketing Manage.*, vol. 20, Jun. 2021, Art. no. 100570.
[34] M. Olmedilla, M. R. Martínez-Torres, and S. Toral, "Prediction and modelling online reviews helpfulness using 1D convolutional neural networks," *Expert Syst. Appl.*, vol. 198, Jul. 2022, Art. no. 116787.
[35] J. Otterbacher, "'Helpfulness' in online communities: A measure of message quality," in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.* Boston, MA, USA: ACM Press, Apr. 2009, pp. 955–964.
[36] A. Ghose and P. G. Ipeirotis, "Estimating the helpfulness and economic impact of product reviews: Mining text and reviewer characteristics," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 10, pp. 1498–1512, Oct. 2011.
[37] 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*, Montreal, QC, Canada, Apr. 2016, pp. 507–517.
[38] H. Fischer, *A History of the Central Limit Theorem*. New York, NY, USA: Springer, 2011.
[39] J. M. Ver Hoef and P. L. Boveng, "Quasi-Poisson vs. negative binomial regression: How should we model overdispersed count data?" *Ecology*, vol. 88, no. 11, pp. 2766–2772, 2007.
[40] S.-H. Wu and J.-W. Wang, "Integrating neural and syntactic features on the helpfulness analysis of the online customer reviews," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining*, Vancouver, BC, Canada, Aug. 2019, pp. 1013–1017.
[41] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org
[42] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn.*, Haifa, Israel, 2010, p. 8.
[43] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. 30th Int. Conf. Mach. Learn.*, vol. 28. Atlanta, GA, USA, 2013, p. 6.
[44] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Math. Program.*, vol. 45, no. 1, pp. 503–528, 1989. [Online]. Available: https://doi.org/10.1007/BF01589116
[45] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval: The Concepts and Technology Behind Search*, 2nd ed. Reading, MA, USA: Addison-Wesley, 2011.
[46] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proc. Conf. EMNLP*. Doha, Qatar: ACL, 2014, pp. 1532–1543.
[47] V. E. Balas, S. S. Roy, D. Sharma, and P. Samui, Eds., *Handbook Deep Learning Applications* (Smart Innovation, Systems and Technologies), vol. 136. Cham, Switzerland: Springer, 2019.
[48] A. Singh and M. Masuku, "Sampling techniques & determination of sample size in applied statistics research: An overview," *Int. J. Econ., Commerce Manage.*, vol. 2, no. 11, pp. 1–22, Aug. 2014.
[49] F. Santosa and W. W. Symes, "Linear inversion of band-limited reflection seismograms," *SIAM J. Sci. Stat. Comput.*, vol. 7, no. 4, pp. 1307–1330, Jul. 1986.
[50] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Statist. Soc. B, Methodol.*, vol. 58, no. 1, pp. 267–288, 1996.
[51] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Ann. Statist.*, vol. 32, no. 2, pp. 407–499, 2004.
[52] Y. Pati, R. Rezaiifar, and P. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proc. 27th Asilomar Conf. Signals, Syst. Comput.*, vol. 1, 1993, pp. 40–44.
[53] G. M. Davis, S. G. Mallat, and Z. Zhang, "Adaptive time-frequency decompositions," *Opt. Eng.*, vol. 33, no. 7, pp. 2183–2191, 1994.
[54] M. Fan, Y. Feng, M. Sun, P. Li, H. Wang, and J. Wang, "Multi-task neural learning architecture for end-to-end identification of helpful reviews," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM)*, Aug. 2018, pp. 343–350.
[55] R. Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, no. 1, pp. 41–75, Jul. 1997.
[56] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 832–844, Aug. 1998.

RISTU SAPTONO (Member, IEEE) received the bachelor's (S.Si.) degree in mathematics from Universitas Sebelas Maret, Surakarta, in 2001, and the master's (M.T.) degree in informatics from the Institut Teknologi Bandung, in 2006. He is currently pursuing the Ph.D. degree with Kyushu University.

His research interests include recommendation systems, statistical-based prediction, and classification.

TSUNENORI MINE received the B.E. degree in computer science and computer engineering and the M.E. and D.E. degrees in information system from Kyushu University, in 1987, 1989, and 1993, respectively.

He is currently an Associate Professor with the Department of Advanced Information Technology, Faculty of Information Science and Electrical Engineering, Kyushu University. His research interests include data mining, text mining, natural language processing, recommendation, and multi-agent systems.