Probability in the Engineering and Informational Sciences (2023), **37:**2 597–620
doi:10.1017/S0269964823000074

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

On hybrid tree-based methods for short-term insurance claims

Zhiyu Quan[1,*], Zhiguo Wang[2], Guojun Gan[2] and Emiliano A. Valdez[2]

[1]Department of Mathematics, University of Illinois at Urbana-Champaign, 1409 W. Green Street (MC-382), Urbana, IL 61801, USA.
[2]Department of Mathematics, University of Connecticut, 341 Mansfield Road, Storrs, CT 06269-1009, USA.
*Corresponding author. E-mail: zquan@illinois.edu

Keywords: Hyperparameter tuning, Pure premium, Regularized regression, Tree-based models, Tweedie generalized linear model

Abstract
Two-part framework and the Tweedie generalized linear model (GLM) have traditionally been used to model loss costs for short-term insurance contracts. For most portfolios of insurance claims, there is typically a large proportion of zero claims that leads to imbalances, resulting in lower prediction accuracy of these traditional approaches. In this article, we propose the use of tree-based methods with a hybrid structure that involves a two-step algorithm as an alternative approach. For example, the first step is the construction of a classification tree to build the probability model for claim frequency. The second step is the application of elastic net regression models at each terminal node from the classification tree to build the distribution models for claim severity. This hybrid structure captures the benefits of tuning hyperparameters at each step of the algorithm; this allows for improved prediction accuracy, and tuning can be performed to meet specific business objectives. An obvious major advantage of this hybrid structure is improved model interpretability. We examine and compare the predictive performance of this hybrid structure relative to the traditional Tweedie GLM using both simulated and real datasets. Our empirical results show that these hybrid tree-based methods produce more accurate and informative predictions.

1. Introduction and motivation

Building regression models for insurance claims poses several challenges. The process can be particularly difficult for individual insurance policies where a large proportion of claims are zeros, and for those policies with a claim, the losses typically exhibit skewness. The degree of skewness in the positive loss distribution varies widely for different types of insurance products. For example, within a class of automobile insurance policies, the losses arising from damages to the property may be medium-tailed but those arising from liability-related claims may be long-tailed. For short-term insurance contracts, the number of claims is referred to as the claim frequency, while the amount of claims, conditional on the occurrence of a claim, is the claim severity. The information that covariates, or explanatory variables, bring to each component in a regression context allows to capture the possible heterogeneity of the individual policies. These two components are used as predictive models for pure premiums, expected loss costs, and aggregate or total claims in an insurance portfolio.

There is a long history of studying insurance claims based on the two-part framework. Several well-known families of distributions for claim frequencies and claim severities are well documented in [15]. As more observed data became increasingly available, additional complexities to traditional regression linear models have been introduced in the literature and in practice; for example, the use of zero-inflated or hurdle models has been discussed to accommodate the shortcomings of traditional models for excess zeros. Interestingly, the Tweedie generalized linear model (GLM) for insurance claims

has been particularly popular because it is adaptable to a mixture of zeros and nonnegative insurance claims. Smyth & Jørgensen [23] described the Tweedie compound Poisson regression models and calibrated such models using a Swedish third-party insurance claims dataset; the models were fitted to examine dispersion within the GLM framework. The Tweedie distribution can be regarded as a reparameterization of the compound Poisson–gamma distribution, so that model calibration can be done within the GLM framework. Xacur & Garrido [25] compared these two approaches and concluded that there is no clear superior method; the paper also discussed the advantages and disadvantages of the two methods. The Tweedie GLM presents a larger pure premium and implies both a higher claim frequency and claim severity due to the constant scale (dispersion) parameter; in other words, the mean increases with the variance. The constant scale parameter also forces an artificial relationship between claim frequency and claim severity. The Tweedie GLM does not have an optimal coefficient, especially under the inflexible distribution assumption, and it leads to a loss of information because it ignores the number of claims. On the other hand, the Tweedie GLM has fewer parameters to estimate and is thus considered more parsimonious than a two-part framework. When insurance claim data contain small losses due to low frequencies, the two-part framework is likely to overlook the internal connection between the low frequency and the subsequent small loss amount. For example, the frequency model often indicates zero number of claims for small claim policies, which leads to a zero loss prediction. Other works related to the Tweedie GLM can be found in [5] and [14].

Within the GLM framework, the claim frequency component may not accurately accommodate imbalances caused by the large proportion of zero claims. Simultaneously, a significant limitation is that the regression structure of the logarithmic mean is restricted to a linear form, which may be too inflexible for real applications. With the rapid expansion of the available data for improved decision-making, there is a growing appetite in the insurance industry to expand its tool kit for data analysis and modeling. However, the industry is unarguably highly regulated, so there is pressure for actuaries and data scientists to provide adequate transparency in modeling techniques and results. To find a balance between modeling flexibility and interpretation, in this paper, we propose a parametric model empowered by tree-based methods with a hybrid structure.

Since [1] introduced the Classification and Regression Tree (CART) algorithm, tree-based models have gained momentum as a powerful machine learning approach to decision-making in several disciplines. In particular, single-tree models, such as the CART algorithm, can be interpreted straightforwardly by visualizing the tree structure. The CART algorithm involves separating the explanatory variable space into several mutually exclusive regions and thus creating a nested hierarchy of branches resembling a tree structure. Each separation or branch is referred to as a node. Each of the bottom nodes of the decision tree, called terminal nodes, has a unique path for observable data to enter the region. Once the decision tree is constructed, it is possible to predict and locate the terminal node for scoring the dataset using a new set of observed explanatory variables.

Before further exploring the hybrid structure, it is worth mentioning that a similar structure called Model Trees (MT), with an M5 algorithm, was first described in [21]. In the M5 algorithm, it constructs tree-based piecewise linear models. While regression trees assign a constant value to the terminal node as the fitted value, MT use a linear regression model at each terminal node to predict the fitted value for observations that reach that terminal node. Regression trees are a special case of MT. Both regression trees and MT employ recursive binary splitting to build a fully grown tree. Thereafter, both algorithms use a cost-complexity pruning procedure to trim the fully grown tree back from each terminal node. The primary difference between regression trees and MT algorithms is that, for the latter step, each terminal node is replaced by a regression model instead of a constant value. The explanatory variables that serve to build those regression models are generally those that participate in the splits within the subtree that will be pruned. In other words, the explanatory variables in each node are located beneath the current terminal node in the fully grown tree.

It has been demonstrated that MT have advantages over regression trees in both model simplicity and prediction accuracy. MT produce decision trees considered not only relatively simple to understand but are also efficient and robust. Additionally, such tree-based methods can exploit local linearity within the

https://doi.org/10.1017/S0269964823000074 Published online by Cambridge University Press

dataset. When prediction performance is compared, it is worth mentioning the difference in the range of predictions produced between traditional regression trees and MT. Accordingly, regression trees are only able to give a prediction within the range of values observed in the training dataset. However, MT are able to extrapolate the prediction range because of the use of the regression models at the terminal node. Especially, when the regression function is a smooth function of explanatory variables, MT may solve the problem that regression trees may not fit the data well. For further discussions of MT and the M5 algorithm, see [22].

Inspired by the structure and advantages of MT when compared to traditional regression trees, we develop a similar algorithmic structure that can uniquely capture the two-part nature of insurance claims. While it is true that MT and regression trees are used for the general regression task, insurance claim datasets present an additional layer of information, the frequency component, which is a classification task. In this paper, we propose hybrid tree-based methods as a two-step algorithm: the first step builds a classification tree to identify the membership of claim occurrence, and the second step uses a regularized regression technique to determine the claim severity at each of the terminal nodes. In essence, such hybrid tree-based methods integrate the peculiarities of both the classification tree and linear regression in the modeling process. These sets of models are suitably described as a hybrid structure, which has the appearance of a segregated two-step structure, but in actuality, they are an integrated approach so that they are more reasonably comparable to the Tweedie GLM than the two-part framework. Furthermore, in model evaluation, it becomes unreasonable to compare the frequency and severity components in segregation.

The rest of this paper has been organized as follows. In Section 2, we provide technical details of the algorithm arising from hybrid tree-based methods applicable to insurance claims, separating the modeling of claim frequency and claim severity. In Section 3, to investigate the predictive accuracy of our hybrid tree-based methods, we create a synthetic dataset based on a simulation produced from a true Tweedie distribution. Within the simulation, we introduced some deviations to mimic real-life situations for a more reasonable comparison. In Section 4, using empirical data drawn from a portfolio of general insurance policies, we illustrate the calibration results and compare the prediction accuracy based on a holdout sample for validation. Section 5 provides visualization of the results of the hybrid structure, which provides a better interpretation. We conclude in Section 6.

2. Description of hybrid tree-based methods

Hybrid tree-based methods use information that arises from both frequency and severity components. As already alluded to in Section 1, it is a two-stage procedure. In the first stage, we construct a classification tree-based model for frequency. In the subsequent stage, we employ a penalized regression model at each terminal node within the tree-based model, based on severity information. Such methods can be drawn from the hybrid structure, which can be viewed as an ecosystem that is able to accommodate a variety of characteristics of datasets.

The mathematical construct of a hybrid tree-based method can be expressed in the following functional form:

$$H(\mathbf{X}, T(\Theta), \boldsymbol{\beta}_{s_m}) = \sum_{m=1}^{M} \underbrace{\beta_{f_m}}_{\text{frequency}} \mathbf{X} \underbrace{\boldsymbol{\beta}_{s_m}}_{\text{severity}} \underbrace{\mathbf{1}_{R_m}(\mathbf{X})}_{\text{risk class}}, \tag{1}$$

where H represents the response of the hybrid tree-based method, given the set of explanatory variables $\mathbf{X}$, the tree structure $T(\Theta)$, and regression model coefficients $\boldsymbol{\beta}_{s_m}$. From Eq. (1), we see that a hybrid structure may also be viewed as piecewise regression models. The tree-based algorithms, $T(\Theta)$, divide the dataset into M subsamples (or risk classes represented by $\mathbf{1}_{R_m}(\mathbf{X})$) and then assign the expected average number of claims β_{f_m}, frequency information. Based on this division, regression models with

https://doi.org/10.1017/S0269964823000074 Published online by Cambridge University Press

coefficients $\boldsymbol{\beta}_{s_m}$ are applied to these subsamples. Each component is further described in the subsequent subsections.

We can determine the type of classification trees that can be adapted in the frequency component according to the information drawn from the dataset. If it records only whether claims were reported or not, we construct a classification tree for a binary response variable. If it records additional information on the number of claims, we construct trees for a count response variable. For binary classification, we can employ some of the most popular algorithms, to list a few: (a) the CART algorithm, (b) efficient algorithm C4.5 [22], with a current updated version C5.0, (c) unbiased approaches like Generalized, Unbiased, Interaction Detection, and Estimation (GUIDE) [17], or (d) Conditional Inference Trees (CTREE) [12]. For the count response variable, we can apply, to list a few, (a) piecewise linear Poisson using the CART algorithm, (b) SUPPORT [2], (c) Poisson regression using GUIDE [16], or (d) Model-Based recursive partitioning (MOB) [26].

After the completion of the classification tree structure, we employ a linear regression model to each of the terminal nodes. The simplest model includes GLM [18] with different distribution models from this family together with regularized regression, such as elastic net regularization [27] among others.

Hybrid tree-based methods build an ecosystem that utilizes modern techniques from both traditional statistics and machine learning. In the following subsections, for illustration purposes, we only show a simple hybrid tree-based method without exploring all possible combinations of the different algorithms suitable at each stage of the procedure. To illustrate, we select the well-known CART algorithm for binary classification of the claim frequency component and least squares regression with elastic net regularization for the claim severity component. Elastic net regularization can perform variable selection and improve prediction accuracy compared to traditional linear regressions without regularization. In order to be self-contained, we illustrate this simple hybrid tree-based method with enough details. Additional details for the severity models are given in Appendix A; see also [13].

2.1. *Claim frequency component*

We denote the response variable as $\mathbf{Y}$, the sample space as $\mathcal{Y}$, and the number of observations as n. The ith observation with p-dimensional explanatory variables is denoted as $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ip})$, $i = 1, \ldots, n$, which is sampled from the space $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_p$. We can separate each claim y_i into y_{i_f}, the claim occurrence or the number of claims, and y_{i_s}, the claim severity or the size of the claim.

In the CART algorithm, a binary classification tree, denoted by $T(\mathbf{x}, \Theta)$, is produced by partitioning the space of the explanatory variables into M disjoint regions $R_1, R_2, \ldots, R_M$ and then assigning a Boolean β_{f_m} for each region R_m, for $m = 1, 2, \ldots, M$. Given a classification tree, each observation can then be classified based on the expression

$$T(\mathbf{x}, \Theta) = \sum_{m=1}^{M} \beta_{f_m} \mathbf{1}_{R_m}(\mathbf{x}), \tag{2}$$

where $\Theta = \{R_m, \beta_{f_m}\}_{m=1}^{M}$ denotes the partition with the assigned Boolean. To be more specific, Boolean $\beta_{f_m} = 1$ when the majority of observations in region R_m have a claim; otherwise it is zero. For other classification tree-based algorithms, especially for the count response variable, β_{f_m} refers to the average number of claims, which can be any integer in $[0, \infty)$, in the region R_m.

The traditional classification loss functions used for classification trees, in Eq. (2), are described as follows:

- Misclassification error: $\text{Mis}(p) = 1 - \max(p, 1 - p)$,
- Gini index: $\text{Gini}(p) = 2p(1 - p)$,
- Cross-entropy or deviance: $\text{Entropy}(p) = -p \log p - (1 - p) \log(1 - p)$,

https://doi.org/10.1017/S0269964823000074 Published online by Cambridge University Press

where p is the proportion of observations in the one class in the node. The default CART algorithm uses the Gini index as a loss function. For multiclass loss functions, see [9].

The regions in the classification tree are determined according to recursive binary splitting. The first step in the splitting process is the discovery of one explanatory variable $x_{\cdot j}$, which best divides the data into two subregions; for example, these regions are the left node $R_{\mathrm{L}}(j,s) = \{\mathbf{x}_i | x_{\cdot j} < s\}$ and the right node $R_{\mathrm{R}}(j,s) = \{\mathbf{x}_i | x_{\cdot j} \geq s\}$ in the case of a continuous explanatory variable. This division is determined as the solution to

$$\underset{j,s}{\operatorname{argmin}}\, w_{\mathrm{L}} \operatorname{Gini}(p_{\mathrm{L}}) + w_{\mathrm{R}} \operatorname{Gini}(p_{\mathrm{R}}), \quad \text{for any } j \text{ and } s, \tag{3}$$

where $w_{\cdot}$ is the weight of the subregion determined by the number of observations split into the subregion divided by the total number of observations before the split, and $p_{\cdot}$ is the proportion of one class in the subregion. Subsequently, the algorithm looks for the next explanatory variable with the best division into two subregions, and this process is applied recursively until meeting some predefined threshold or reaching a minimum size of observations at the terminal node.

To control for model complexity, we can use *cost-complexity pruning* to trim the fully grown tree T_0. We define the loss in region R_m by

$$L_m(T) = w_m \operatorname{Mis}(p_m),$$

where $\operatorname{Mis}(p_m)$ is the misclassification error. For any subtree $T \subset T_0$, we denote the number of terminal nodes in this subtree by $|T|$. To control the number of terminal nodes, we introduce the tuning hyperparameter $\tau \geq 0$ to the loss function by defining the new cost function as

$$C_\tau(T) = \sum_{m=1}^{|T|} L_m(T) + \tau |T|. \tag{4}$$

Clearly, according to this cost function, the tuning hyperparameter penalizes a large number of terminal nodes. The idea then is to find the subtrees $T_\tau \subset T_0$ for each τ and choose the subtree that minimizes $C_\tau(T)$ in Eq. (4). Furthermore, the tuning hyperparameter τ governs the tradeoff between the size of the tree (model complexity) and its goodness-of-fit to the data. Large values of τ result in smaller trees (simple model), and as the notation suggests, $\tau = 0$ leads to the fully grown tree T_0. Additional hyperparameter tuning is performed to control the tree depth through *maxdepth*, which is the maximum depth of any node in the final tree. Through *cost-complexity pruning* and controlling *maxdepth*, the size of the classification tree is constrained, which guarantees a sufficient sample size for the terminal nodes to build a linear model for the severity component. Furthermore, it is feasible to extend the hyperparameter setting with an alternative choice of loss function described above.

2.2. Claim severity component

After building the classification tree structure, we next apply a linear model to the terminal nodes for the claim severity component. When controlling for model complexity, we include a threshold as an additional hyperparameter to determine if we should build a linear model or directly assign zero to the terminal node. For example, if *zeroThreshold* is 80%, then we should directly assign zero to the terminal nodes that contain more than 80% of zero claims. Furthermore, if the terminal node contains less than a certain number of observations, say 40, which can also be determined by hyperparameter tuning, we can directly use the average as the prediction, which is similar to regression trees. Otherwise, we need to build a linear model to the terminal nodes. While any member of the GLM exponential family is applicable for continuous claims, we find that the special case of GLM with the Gaussian family, or just ordinary linear regression, is sufficiently suitable for our purposes. In Section 5, we will

https://doi.org/10.1017/S0269964823000074 Published online by Cambridge University Press

empirically discuss the rationality of using the Gaussian family, even in insurance losses that typically exhibit skewness.

At each terminal node R_m, the linear coefficient $\boldsymbol{\beta}_{s_m} = (\beta_{s_m0}, \beta_{s_m1}, \ldots, \beta_{s_mp})^{\mathrm{T}}$ can be determined based on elastic net regularization:

$$\widehat{\boldsymbol{\beta}}_{s_m} = \underset{\boldsymbol{\beta}_{s_m}}{\operatorname{argmin}} \frac{1}{n_m} \sum_{i=1}^{n_m} \ell\left(y_i, \beta_{s_m0} + \sum_{j=1}^{p} x_{ij}\beta_{s_mj}\right) + \lambda \left[(1-\alpha)\frac{1}{2}\sum_{j=1}^{p}\beta_{s_mj}^2 + \alpha \sum_{j=1}^{p}|\beta_{s_mj}|\right], \tag{5}$$

where n_m is the number of sample in the terminal node, α controls the elastic net penalty and bridges the gap between Least Absolute Shrinkage and Selection Operator (LASSO) (when $\alpha = 1$) and ridge regression (when $\alpha = 0$). Zou & Hastie [27] pointed out that LASSO has few drawbacks. For example, when the number of explanatory variables p is larger than the number of observations n, LASSO selects at most n explanatory variables. Additionally, if there is a group of pairwise correlated explanatory variables, LASSO randomly selects only one explanatory variable from this group and ignores the rest. Zou & Hastie [27] also empirically showed that LASSO has a subprime prediction performance compared to ridge regression. Therefore, elastic net regularization is proposed, which uses a convex combination of ridge and LASSO regression. Elastic net regularization is able to better handle correlated explanatory variables and perform variable selection and coefficient shrinkage.

If explanatory variables are correlated in groups, an α around 0.5 tends to select the groups in or out altogether. For regularized least squares with the elastic net penalty, this reduces to

$$\widehat{\boldsymbol{\beta}}_{s_m} = \underset{\boldsymbol{\beta}_{s_m}}{\operatorname{argmin}} \frac{1}{2n_m} \sum_{i=1}^{n_m} \left(y_i - \beta_{s_m0} - \sum_{j=1}^{p} x_{ij}\beta_{s_mj}\right)^2 + \lambda \left[(1-\alpha)\frac{1}{2}\sum_{j=1}^{p}\beta_{s_mj}^2 + \alpha \sum_{j=1}^{p}|\beta_{s_mj}|\right].$$

This problem can also be solved using the coordinate descent algorithm. We omit the detailed solution, which is a similar process to solve LASSO using the coordinate descent algorithm described in Appendix A. However, it is worth mentioning that the expression for $\widehat{\beta}_{s_mk}$ can be numerically estimated as follows:

$$\widehat{\beta}_{s_mk} = \frac{1}{1+\lambda(1-\alpha)} \left(\frac{\sum_{i=1}^{n_m}|\tilde{y}_i x_{ik}|}{\sum_{i=1}^{n_m} x_{ik}^2} - \frac{\lambda\alpha}{\frac{1}{n_m}\sum_{i=1}^{n_m} x_{ik}^2}\right)^{+} \operatorname{sgn}\left(\sum_{i=1}^{n_m} \tilde{y}_i x_{ik}\right).$$

The mathematical foundation for elastic net regularization can be found in [3]. It should also be noted that all the regularization schemes mentioned previously have simple Bayesian interpretations; see [9] and [13]. A review paper for statistical learning applications in general insurance can be found in [19].

Based on hyperparameters λ and α, the linear structure varies among ordinary linear regression (when $\lambda = 0$), ridge regression, LASSO, and elastic net regularization. In addition, hybrid tree-based methods perform automatic variable selections for claim severity, which is distinctively different from MT mentioned in Section 1. MT consider only explanatory variables that are not used in the splitting process involved in the tree structure as candidates when it builds models to the terminal nodes. The reason for elastic net is to amplify the information remaining after the classification step. This is partly because some explanatory variables may be significant for the claim frequency component but not for the severity component. For this reason, the practice of choosing the correct set of explanatory variables is important, and thereby, we employ elastic net. In the extreme case where, for example, gender is an explanatory variable, only observations of one particular gender may remain in the terminal node, then it becomes redundant to consider gender as an explanatory variable.

Previously, we described hybrid tree-based methods as a two-step algorithm that appears to be very similar to a traditional two-part framework. However, there is a significant difference between them. In the traditional two-part framework, we build the best model based only on information from claim

https://doi.org/10.1017/S0269964823000074 Published online by Cambridge University Press

frequency at the beginning, and this claim frequency model will be static in the following process. In fact, there are two optimization processes, one for claim frequency and the other for claim severity. On the other hand, hybrid tree-based methods have only a single optimization process. Based on the hyperparameter settings, the structure of the classification tree, or the model of the claim frequency component, can be very flexible. In this sense, hybrid tree-based methods are more similar to the Tweedie GLM. This is why it is most suitable to compare our hybrid tree-based methods with the Tweedie GLM in the subsequent discussions.

Algorithm 1 summarizes the details of implementing hybrid tree-based methods based on the `rpart` and `glmnetUtils` packages in R. Although the details have been necessarily omitted, we note that all model hyperparameters considered in the rest of the paper have been properly tuned based on grid search.

Algorithm 1: Implementation of hybrid tree-based methods

Input: Formula for the tree, formula for the linear model, training dataset $\mathbf{x}$, cost-complexity parameter cp, maximum depth of final tree $maxdepth$, the threshold for the proportion of zeros in the node to be considered to build a linear model $zeroTreshold$, the choice of elastic net regularization $glmWhich$, penalty size $glmLambda$.

Output: Tree structure, linear models at each node, node information, $glmWhich$, $glmLambda$, fitted dataset.

1 Grow a tree on a training dataset using recursive binary splitting. Use the stopping criterion $maxdepth$;
2 Prune the tree using cost-complexity pruning with cp;
3 Assign node information to each observation;
4 **for** *each terminal node* **do**
5 If zero claim exceeds the $zeroTreshold$ in the node, then we assign zero as the prediction;
6 Else, we build a linear model using observations in this node. If the number of observations in this node is smaller than 40 (a hyperparameter that can be tuned), then we assign the average of the response variable as the prediction; otherwise, we fit ordinary least squares with elastic net regularization model;
7 **end**
8 **Return:** Tree structure, linear models at each node, node information, $glmWhich$, $glmLambda$, fitted dataset;

Figure 1 provides a simplified illustration of the underlying procedure in a hybrid tree-based method. In particular, it assumes two tree depths and observes that during the splitting process, the combination of zeros and nonzeros can vary at different nodes. When all remaining observations in the final node are zeros, the predicted value is zero. When there is a more balanced combination of zeros and nonzeros, elastic net regularized models are fitted. In the case where there are not enough observations, the reasonable prediction is the average of all the response values of the observations in the final node. In the implementation of hybrid tree-based methods, the maximum tree depth, cost-complexity parameter, the thresholds for the proportion of zeros in the node, and minimum observations for building regression are hyperparameters to be tuned.

3. Simulation study

In this section, we conduct a simulation study to assess the performance of a simple hybrid tree-based method proposed in this paper relative to that of the Tweedie GLM. The synthetic dataset has been optimally created to replicate a real dataset. This synthetic dataset contains continuous explanatory variables and categorical explanatory variables with a continuous response variable sampled from a

https://doi.org/10.1017/S0269964823000074 Published online by Cambridge University Press

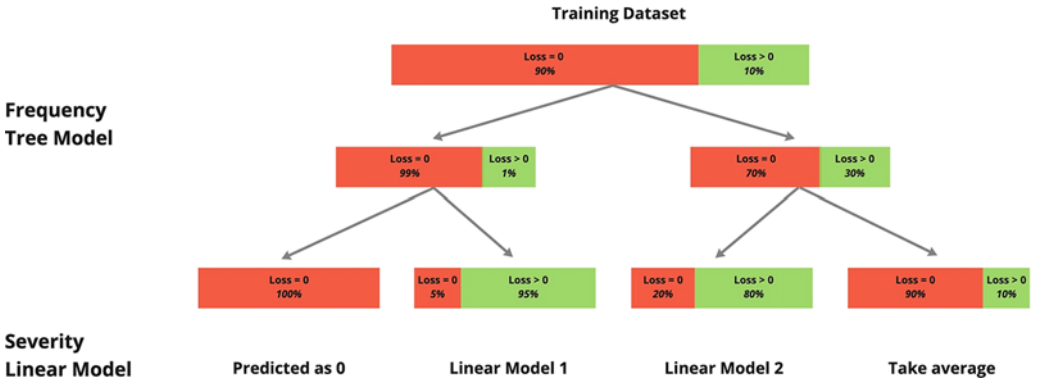

Figure 1. *A simplified illustration of hybrid tree-based methods.*

Tweedie-like distribution. In some sense, the observed values of the response variable behave like a Tweedie, but some noise was introduced to the data for a fair comparison.

We begin with the continuous explanatory variables by generating sample observations from the multivariate normal with covariance structure

$$\mathbf{x} \sim N_p(0, \Sigma),$$

where $Cov(x)_{ij} = (0.5)^{i-j}$. It is possible to increase the correlation to a value much higher than 0.5; however, it would cause too much unnecessary noise and therefore leads to an unfair comparison. Indeed, we tested different values such as 0.8 and it favored the hybrid structure. Continuous explanatory variables close to each other has higher correlations, and the correlation effect diminishes with increasing distance between two explanatory variables. In real datasets, we can easily observe correlated explanatory variables among hundreds of possible features. It is also very challenging to detect the correlation between explanatory variables when an insurance company merges with third-party datasets. Multicollinearity reduces the precision of the estimated coefficients of correlated explanatory variables. It can be problematic to use GLM without handling multicollinearity in the real dataset.

Categorical explanatory variables are generated by random sampling from the set of integers $(-3, -2, 1, 4)$ with equal probabilities. Categorical explanatory variables do not have any correlation structure among them.

We create $10,000$ observations (with 94.31% zero claims) with 60 explanatory variables, including 10 continuous explanatory variables with relatively larger coefficients, 10 continuous explanatory variables with relatively smaller coefficients, 10 continuous explanatory variables with coefficients of 0, which means no relevance, 10 categorical explanatory variables with relatively larger coefficients, 10 categorical explanatory variables with relatively smaller coefficients, and 10 categorical explanatory variables with the coefficient 0, which similarly implies no relevance. In effect, there are three groups of explanatory variables according to the size of the linear coefficients. In other words, there are strong signal explanatory variables, weak signal explanatory variables, and noise explanatory variables. Here are the true linear coefficients $\boldsymbol{\beta}_{\mathbf{Poisson}}$ and $\boldsymbol{\beta}_{\mathbf{Gamma}}$ used:

$$\boldsymbol{\beta}_{\mathbf{Poisson}} = (-0.1, \underbrace{0.5, \ldots, 0.5}_{\text{10 continuous}}, \underbrace{0.1, \ldots, 0.1}_{\text{10 continuous}}, \underbrace{0, \ldots, 0}_{\text{10 continuous}}, \underbrace{-0.5, \ldots, -0.5}_{\text{10 categorical}}, \underbrace{0.1, \ldots, 0.1}_{\text{10 categorical}}, \underbrace{0, \ldots, 0}_{\text{10 categorical}})^{\mathrm{T}},$$

$$\boldsymbol{\beta}_{\mathbf{Gamma}} = (6, \underbrace{0.5, \ldots, 0.5}_{\text{10 continuous}}, \underbrace{-0.1, \ldots, -0.1}_{\text{10 continuous}}, \underbrace{0, \ldots, 0}_{\text{10 continuous}}, \underbrace{0.5, \ldots, 0.5}_{\text{10 categorical}}, \underbrace{-0.1, \ldots, -0.1}_{\text{10 categorical}}, \underbrace{0, \ldots, 0}_{\text{10 categorical}})^{\mathrm{T}}.$$

https://doi.org/10.1017/S0269964823000074 Published online by Cambridge University Press

The first coefficients refer to the intercepts. Because of the equivalence of the compound Poisson–gamma with the Tweedie distributions, we generated samples drawn from a Poisson distribution for the claim frequency and a gamma distribution for the claim severity.

To illustrate possible drawbacks of using a Tweedie GLM, we use different linear coefficients for the frequency part and the severity part. The absolute value of the linear coefficients is the same but with a different sign. In real life, there is no guarantee that the explanatory variables have equal coefficients in both frequency and severity parts. The response variable generated by the compound Poisson–gamma distribution framework using the modified `rtweedie` function in the `tweedie` R package with frequency and severity as components is presented as follows:

Frequency part: Poisson distribution with a log link function:

$$N_i \sim \text{Poisson}(\lambda) \text{ with } \lambda = \lambda_i \text{ for given } x_i,$$

where $\lambda_i = \dfrac{\exp(\mathbf{x}_i\boldsymbol{\beta}_{\mathbf{Poisson}})^{2-\text{power}}}{\phi \times (2-\text{power})}$, with power = 1.5 *and* $\phi = 2$.

Severity part: Gamma distribution with a log link function:

$$Y_j \sim \text{Gamma}(\alpha, \beta) \text{ with } \alpha = \frac{2-\text{power}}{\text{power}-1} \text{ and } \beta = \beta_i \text{ for given } x_i,$$

where $\beta_i = \dfrac{\exp(\mathbf{x}_i\boldsymbol{\beta}_{\mathbf{Gamma}})^{1-\text{power}}}{\phi \times (\text{power}-1)}$.

Response variable:

$$Y_i^r = \sum_{j=1}^{N_i} Y_j \text{ for given } x_i,$$

where the superscript r has been used to distinguish the response and the gamma variables.

We introduce a variety of noise to the true Tweedie model in the following ways: possible multicollinearity (weak correlations) among continuous variables, zero coefficients for nonrelevant explanatory variables, different coefficients (same size but different sign) for frequency and severity components, and additional white noise to positive claims. These common noises are what make the real-life data deviate from the true model in practice. It makes for an impartial dataset that can provide for a fair comparison.

Once we have produced the simulated (or synthetic) dataset, we apply both the Tweedie GLM and hybrid tree-based methods to predict values of the response variables, given the available set of explanatory variables. The prediction performance between the two models is then compared using several validation measures as summarized in Table 1. In Appendix B, Table A1 describes details for validation measures. In many cases, machine learning models often outperform traditional statistical techniques regarding a single validation measure. This is especially true when machine learning models are optimized regarding that validation measure. To avoid choosing only the most beneficial results, we include more validation measures in the following study in order to draw a stronger and more comprehensive conclusion.

From Table 1, we find that the Tweedie GLM's fitting performance is worse even under the true model assumption, albeit in the presence of noise. As we described in Section 2, after hyperparameter tuning, hybrid tree-based methods with elastic net regression are able to automatically perform coefficient shrinkage via the L2 norm and variable selection based on the tree-structure algorithm and L1 norm. In a simulation setting that mimics real life, hybrid tree-based methods provide more promising prediction results than the widely accepted Tweedie GLM.

https://doi.org/10.1017/S0269964823000074 Published online by Cambridge University Press

Table 1. *Model performance on the simulation dataset.*

Validation measure	Gini	R^2	CCC	RMSE	MAE
Tweedie GLM	*0.95*	−1.56	*0.49*	1.55	0.07
Hybrid tree-based method	0.84	*0.15*	0.27	*0.89*	*0.06*

CCC, concordance correlation coefficient.

Table 2. *Model performance on the simulation dataset without nonrelevant explanatory variables.*

Validation measure	Gini	R^2	CCC	RMSE	MAE
Tweedie GLM	*0.94*	−2.71	*0.28*	2.07	0.15
Hybrid tree-based method	0.86	*0.13*	0.23	*1.00*	*0.09*

CCC, concordance correlation coefficient.

To rule out unfairness caused by improper variable selection, we reconstruct the simulation dataset without the nonrelevant explanatory variables. Therefore, the only difference from the previous simulated dataset is that we have 40 explanatory variables instead of 60. Using the same procedure, we obtain the model performance shown in Table 2. We can observe consistent results. Three out of five validation measures show that hybrid tree-based methods outperform the Tweedie GLM. However, the two weaker results are comparable.

In summary, we used a Poisson–gamma distribution within a two-part framework, adding some noise that mimics real-life situations to create the simulated datasets. Then, we compared the model accuracy between the Tweedie GLM and hybrid tree-based methods. We conclude that even under the linear assumption, hybrid tree-based methods can perform better than the Tweedie GLM.

4. Empirical application

We compare the Tweedie GLM with hybrid tree-based methods using one coverage group from the Wisconsin Local Government Property Insurance Fund (LGPIF) data. This dataset comes from a project at the University of Wisconsin's actuarial research group, and further details about this dataset can be found on their website.[1] The coverage group is called Building and Contents (BC), which provides insurance for buildings and properties of local government entities including counties, cities, towns, villages, school districts, and other miscellaneous entities. We draw observations from the years 2006 to 2010 as the training dataset and the year 2011 as the test dataset. Tables 3 and 4 summarize the training dataset and the test dataset, respectively.

4.1. Conventional techniques

To examine and compare model performance, we include three conventional techniques in our analysis.

We replicate the Tweedie GLM as noted in [6]. The linear coefficients for the Tweedie GLM have been provided in Table A11 in [6]. It was pointed out that the Tweedie GLM may not be ideal for the dataset after visualizing the cumulative density function plot of jittered aggregate losses, as depicted in Figure A6 of [6]. Despite its shortcomings, the Tweedie GLM is used primarily for comparison purposes.

In addition, to compare the difference between hybrid and traditional tree-based methods, we apply the CART algorithm, the regression tree, to LGPIF data. We perform a grid search on the training dataset with 10-fold cross-validation. The final model has hyperparameters set with a minimum of

[1] https://sites.google.com/a/wisc.edu/jed-frees/

https://doi.org/10.1017/S0269964823000074 Published online by Cambridge University Press

Table 3. *Summary statistics of the training dataset, 2006–2010.*

Response variables	Description	Minimum	1st Q	Mean	Median	3rd Q	Maximum
ClaimBC	BC claim amount in million	0	0	0.02	0	0	12.92
Continuous variables							
CoverageBC	Log of BC coverage amount	−7.95	0.78	2.12	2.42	3.6	7.8
lnDeductBC	Log of BC deductible level	6.21	6.21	7.15	6.91	7.82	11.51
Categorical variables							Proportions
NoClaimCreditBC	Indicator for no BC claims in previous year						32.93%
TypeCity	EntityType: City						14.03%
TypeCounty	EntityType: County						5.80%
TypeMisc	EntityType: Miscellaneous						10.81%
TypeSchool	EntityType: School						28.25%
TypeTown	EntityType: Town						17.33%
TypeVillage	EntityType: Village						23.78%

Table 4. *Summary statistics of the test dataset, 2011.*

Response variables	Description	Minimum	1st Q	Mean	Median	3rd Q	Maximum
ClaimBC	BC claim amount in million	0	0	0.02	0	0	2.75
Continuous variables							
CoverageBC	Log of BC coverage amount	−7.84	0.91	2.27	2.56	3.74	7.78
lnDeductBC	Log of BC deductible level	6.21	6.21	7.24	6.91	7.82	11.51
Categorical variables							Proportions
NoClaimCreditBC	Indicator for no BC claims in previous year						50.87%
TypeCity	EntityType: City						14.06%
TypeCounty	EntityType: County						6.48%
TypeMisc	EntityType: Miscellaneous						11.42%
TypeSchool	EntityType: School						27.67%
TypeTown	EntityType: Town						16.53%
TypeVillage	EntityType: Village						23.84%

eight observations in a region to attempt the recursive binary split, and the cost-complexity pruning parameter (*cp*) is 0.05. It is worth mentioning that the regression tree uses the simple average at the terminal nodes as the prediction.

We also employ the deep neural network (DNN) model to compare prediction performance. Bayesian optimization is employed for hyperparameter tuning and neural network architecture calibration. The final neural network model consists of three hidden layers, which have 284, 156, and 352 nodes and dropout rates of 0.44, 0.49, and 0.11, respectively, each followed by the LeakyReLu activation function.

https://doi.org/10.1017/S0269964823000074 Published online by Cambridge University Press

4.2. Calibration results of hybrid tree-based methods

We employ two hybrid tree-based methods for this empirical application. For binary classification, we use the CART algorithm, and at the terminal node, we use either GLM with the Gaussian family or elastic net regression with the Gaussian family. Let us denote these two types of hybrid tree-based methods, respectively, as HTGlm and HTGlmnet.

For the model calibration, we perform a grid search on the training dataset with 10-fold cross-validation to find the optimal set of hyperparameter values. HTGlm has been built based on the following hyperparameter set: eight minimum number of observations in a region to attempt the recursive binary split, the cost-complexity pruning parameter (*cp*) is 0.0001, maximum depth of the tree (*maxdepth*) is 8, and the threshold for the percentage of zeros in the node to determine whether to build the linear model or assign *zero* as the prediction (*zeroThreshold*) is 0.25. At first glance, the tuned hyperparameter setting indicates that a fairly deep tree comes with a low *zeroThreshold*, which is somewhat surprising. However, this is a purely data-driven result of optimizing the root mean square error (RMSE) objective loss function. The other hybrid tree-based method, HTGlmnet, has the following hyperparameter settings: minimum observations in a region for splitting is 8, *cp* is 0.0002, *maxdepth* is 10, *zeroThreshold* is 0.23, the balance between L1 norm and L2 norm (*glmWhich*) is 1, and the size of the regularization parameter by the best cross-validation result (*glmLambda*) is *lambda.min*.

In regard to the calibration of hybrid tree-based methods, it might be helpful to point out a few observations. The size of tree-based models is controlled by *maxdepth* and *cp*, which constrain the growing tree and prune it when it has grown too large. Apparently, growing a large tree with a small dataset is not ideal. At the terminal node, *zeroThreshold* has another function that guards against mistakes caused by the frequency part. Put another way, if we do not feel comfortable with some positive claims allocated with a certain degree of the proportion of zeros by the tree structure, we can increase the *zeroThreshold* and create more linear models at the terminal nodes instead of assigning the zero directly. However, this results in higher computation costs. Furthermore, we find that there could be multiple hyperparameter settings that lead to the optimal solution. Obviously, there are many other options to implement the optimization process. For example, use the mean absolute error (MAE) as a validation metric, and employ Bayesian optimization instead of the grid search when we have limited computation resources.

4.3. Model performance

We use several validation measures to examine the performance of different models, similar to what was done in the simulation. To make for an easier comparison, we added the heat map in addition to the prediction accuracy table based on various validation measures. We present results separately for training and test datasets. A rescaled value of 100 (colored blue) indicates that it is the best model among the comparisons; on the other hand, a rescaled value of 0 (colored red) indicates that it is the worst. For further details on this heat map for validation, see [20].

From Figure 2(a), we find that the Tweedie GLM has the worst performance of model fit in general and the CART algorithm and HTGlm have better performance in model fitting. From Figure 2(b), we find that HTGlm has the best performance in the test dataset. It also shows that hybrid tree-based methods prevent overfitting compared to traditional regression trees using the CART algorithm. One possible reason is that the hybrid structure extrapolates the prediction range by using regression models at the terminal node. In general, hybrid tree-based methods perform consistently well for both training and test datasets. Because hybrid tree-based methods are considered piecewise linear models, the model generalization is inherited from the linear model. We can also note that HTGlmnet has a poorer performance compared to HTGlm, for both training and test datasets. The advantage of regularization does not demonstrate significant effects on the LGPIF dataset because of the limited number of explanatory variables. Furthermore, we observe a similar phenomenon as the CART algorithm also performs quite well compared to HTGlmnet, which is also likely a reflection of the fact that there are very few explanatory variables. However, in the simulation study, we showed the advantage of regularization because the data

https://doi.org/10.1017/S0269964823000074 Published online by Cambridge University Press

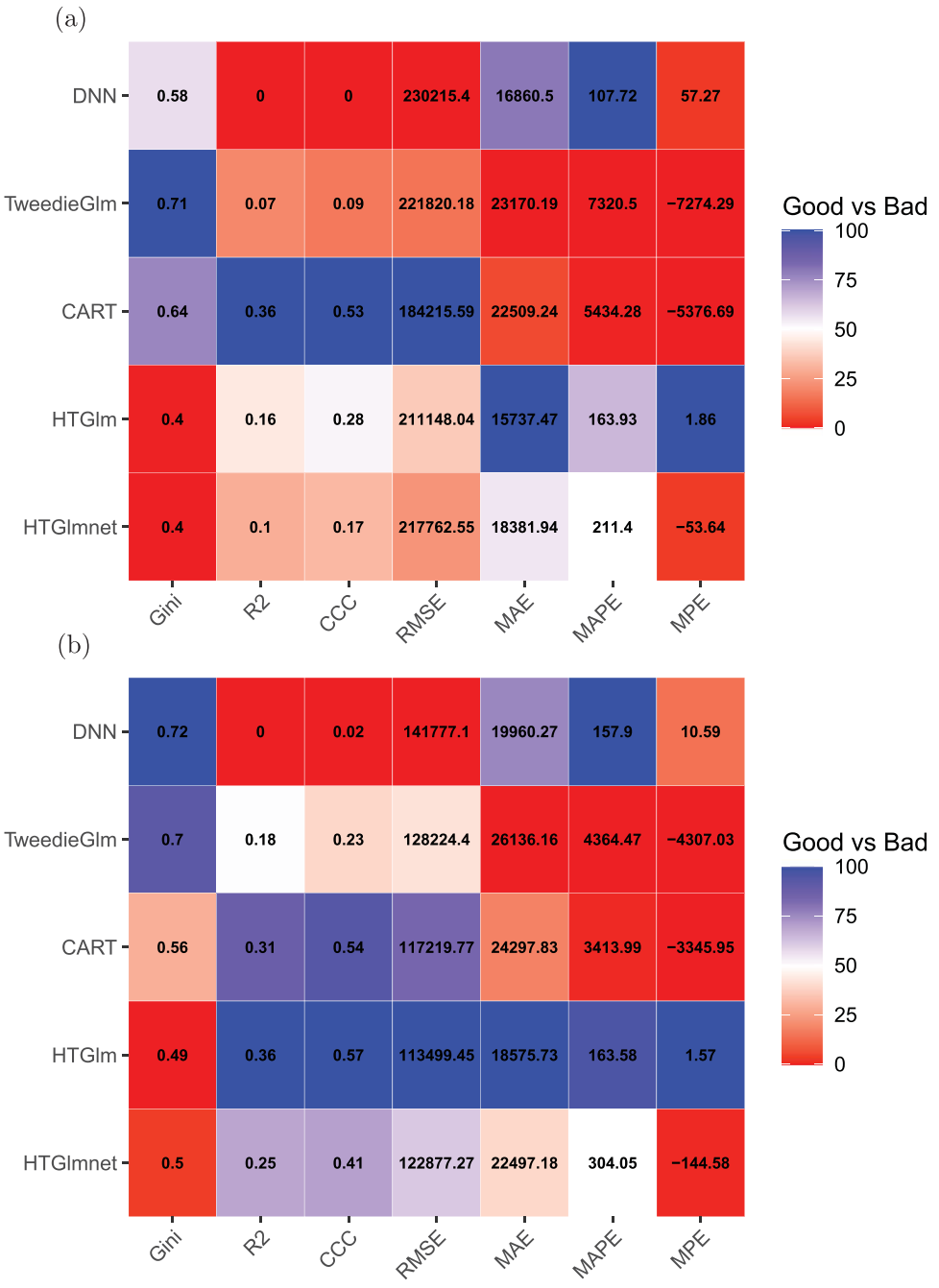

Figure 2. *Heat maps of model comparison based on various validation measures. (a) Model performance based on training dataset. (b) Model performance based on test dataset.*

has a more significant number of explanatory variables. The simulated data may be a better replication of real-life insurance loss compared to LGPIF data with limited explanatory variables. It is also worth mentioning the discrepancy in the model performance regarding the Gini Index. Based on the definition of the Gini Index, the calculation involves ranking the predicted value. The Tweedie GLM and DNN predictions do not include a large number of zeros; however, in the hybrid structure, we assign zero

https://doi.org/10.1017/S0269964823000074 Published online by Cambridge University Press

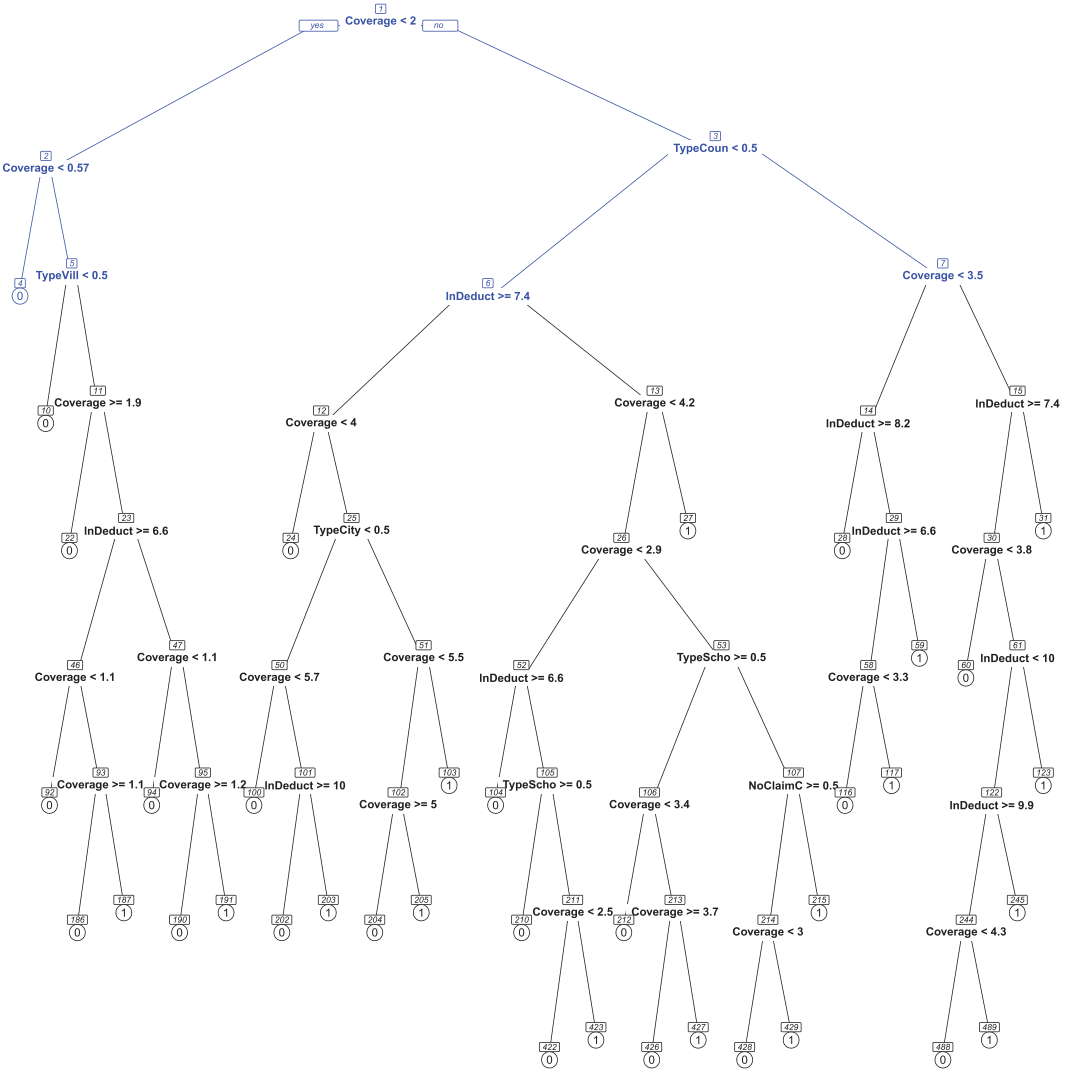

Figure 3. *Tree paths with highlighted nodes.*

prediction to a certain terminal node with a high percentage of zero claims. Consequently, the hybrid structure has an inferior performance regarding the Gini Index due to the uncertainty of ranking zero predictions. The DNN provides one aspect of the black-box model's prediction performance. LGPIF dataset is not large enough for the DNN to observe the deep double descent phenomenon. In summary, hybrid tree-based methods outperform the Tweedie GLM based on various validation measures.

5. Visualization and interpretation

The frequency component of hybrid tree-based methods can clearly be visualized as a tree structure like a simple CART algorithm. Here, we use HTGlm to illustrate the visualization and interpretation of hybrid tree-based methods. In Appendix C, Figure A1 shows the classification tree model for the frequency. We can follow the tree path and construct the density plot for the response variable for each node. This process will show a straightforward visualization of how the sample changes with every split. Figure 3 shows the classification tree, and we highlighted (blue) nodes two depths away from the root.

https://doi.org/10.1017/S0269964823000074 Published online by Cambridge University Press

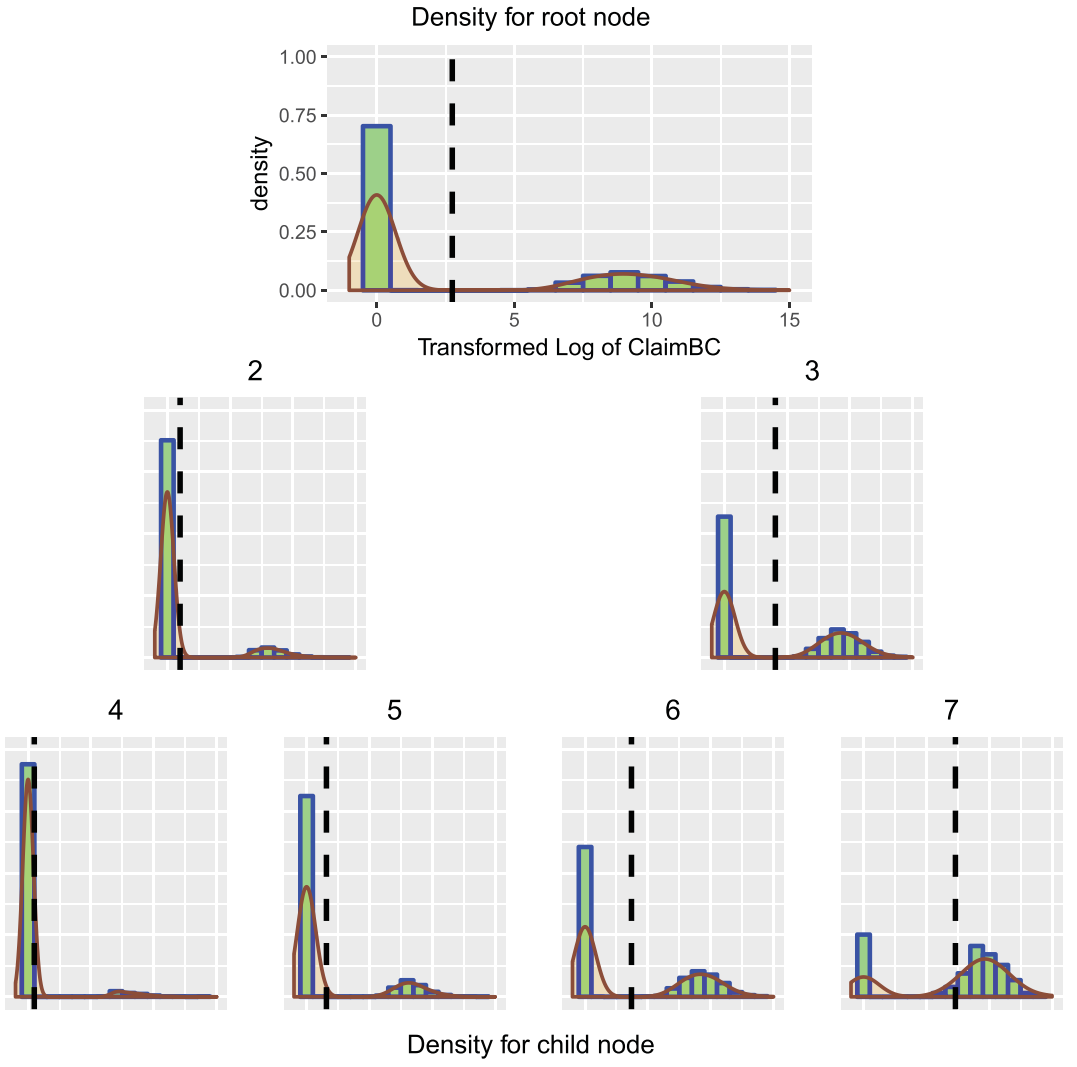

Figure 4. *Classification tree for the frequency.*

Consequently, we have seven nodes including the root nodes. Here we denote the node with numbers for the purpose of identification. Among the highlighted nodes, node four (4), on the left, is the terminal node and the others are intermediate nodes. Figure 4 shows the density of the response variable for the seven nodes mentioned in Figure 3. For better visualization, Figure 4 presents the logarithm of one plus the claim amounts instead of the actual claim amounts. The root node has zero point mass, as expected, and the black dashed line is the mean of the response variable at the node. After the first split using CoverageBC, the mean shifted toward a different direction, and the percentage of the point mass at zero also changed in a different direction. At the fourth terminal node, we can barely see any positive claims with the mean shifting toward zero, and this terminal node ultimately assigns zero as the prediction. We can see that the classification tree algorithm, done by recursive binary splitting, divides the dataset into subspaces that contain more similar observations. The same phenomenon can also be observed in the CART algorithm, which uses a tree structure to separate similar data to the same group in the terminal node and then uses mean or mode as the predicted value. For the same reason, the proposed simple hybrid tree-based method applies linear models with the Gaussian family in the terminal node without setting the choice of distribution as a tuning parameter. After dividing the space of explanatory

https://doi.org/10.1017/S0269964823000074 Published online by Cambridge University Press

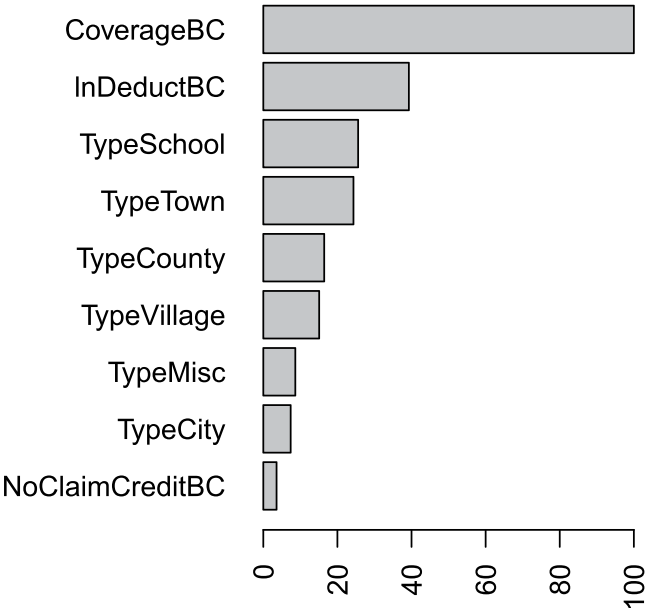

Figure 5. *Variable importance for the claim frequency.*

variables into subspaces, it may be more appropriate to make distribution assumptions and apply the linear model to the subsamples. In a sense, hybrid tree-based methods employ a "divide and conquer" algorithm, which may be a promising solution for imbalanced datasets.

Figure 3 shows the path to terminal nodes, and we can see a zero or a one, where a linear model is then fitted, for each terminal node. The structure and path of the tree can provide an indication of the risk factors based on the split of the explanatory variables. Figure 5 shows the variable importance calculated from the classification tree model for the frequency part. Coverage and deductible information are much more important than location and previous claim history to identify whether the claim occurred or not. It is not surprising that the deductible has an important impact on claim frequency, as a claim occurrence will occur only if the amount exceeds the deductible. At each terminal node, we can also extract the regression coefficients and interpret the estimated linear regression model. Table 5 provides the regression coefficients at each of the nonzero terminal nodes. The node number has the same label as in Figure 3. For example, the terminal node 245 can be reached as follows: $CoverageBC \geq 2$, $TypeCounty$, $Coverage \geq 3.5$, $lnDeductBC \geq 7.4$, $CoverageBC \geq 3.8$, $lnDeductBC < 10$, and $lnDeductBC < 9.9$, and we estimated the linear model with regression coefficients: $Intercept = -172,854$, $CoverageBC = 15,251$, $lnDeductBC = 16,777$, and $NoClaimCreditBC = -16,254$. Similarly, Eq. (1) can be specified based on the tree structure and regression coefficients. Hence, we are able to write out the hybrid tree-based method in the piecewise linear form.

At the terminal node, the linear model is built that is based on the regularized algorithm and is based solely on the accuracy of the prediction from cross-validation. It does not rely on statistical inference, and indeed, some regression coefficients are not statistically significant. However, we can tune hybrid tree-based methods by balancing prediction accuracy and statistical inference. For example, when we try to measure the impact of a specific explanatory variable on the claim severity, the coefficient should be statistically significant. To find the reliable coefficient, we might attempt to merge some child nodes or split the dataset further and then build another regression model. It is possible that these attempts may not yield optimal prediction performance due to the fact that statistical inference is prioritized over prediction accuracy.

The primary advantage of hybrid tree-based methods is its intuitive interpretability. It is possible to extend our approach that considers ensemble techniques such as boosting or bagging using hybrid

https://doi.org/10.1017/S0269964823000074 Published online by Cambridge University Press

Table 5. *Regression coefficients at the terminal nodes.*

	Estimates						
Terminal node	245	31	123	27	203	103	187
(Intercept)	−172,854	24,263	7,922,014	150,264	−1,907,177	155,878	−303,658
TypeCity				17,218			
TypeCounty							
TypeMisc					62,714		
TypeSchool							
TypeTown							
CoverageBC	15,251	26,601	1,102,629	9,454	447,654	2,772	287,021
lnDeductBC	16,777	−15,412	−1,232,514	−25,246	−70,542	−1,526	
NoClaimCreditBC	−16,254	−34,160		−21,825	−67,030	−13,229	−13,632

tree-based methods as base learners. Such extensions can significantly improve prediction accuracy, however, sacrificing model interpretability.

Furthermore, hybrid tree-based methods may capture the dependence between frequency and severity. Based on the hybrid structure, the severity model is conditionally built on tree-based partitions created using frequency information. Hence, it inherently allows for a certain dependence between frequency and severity. We observe this phenomenon in Table 5. Most regression coefficients at the terminal nodes do not include location information, which might already be incorporated into the tree-based model. With this in mind, the proposed hybrid tree-based method has an added advantage over the traditional Tweedie GLM, which assumes that frequency and severity are independent under the compound Poisson–gamma framework.

6. Conclusions

With the development of data science and increased computational capacity, insurance companies can benefit from existing large datasets collected over a stretched period of time. On the other hand, it is also challenging to build a model on "big data." It is especially true for modeling claims of short-term insurance contracts since the dataset presents many unique characteristics. For example, claim datasets typically have a large proportion of zero claims that lead to imbalances in classification problems, a large number of explanatory variables with high correlations among continuous variables, and many missing values in the datasets. It can be problematic if we directly use traditional approaches such as the Tweedie GLM without additional modifications to these characteristics. Hence, it is often not surprising to find a less desirable prediction accuracy of conventional methods for real-life datasets.

In this paper, to fully capture the information available in large insurance datasets, we propose the use of hybrid tree-based methods as learning machinery and an expansion to the available set of actuarial toolkits for claim prediction. We have reiterated the many benefits that can be derived from such a hybrid structure. The use of classification trees to model the frequency component has the advantages of being able to handle imbalanced classes and to naturally subdivide portfolios into risk classes, as well as the many benefits of tree-based models; see also [10, 20]. The use of a regression model for the severity component provides the flexibility of identifying factors that are important predictors for accuracy and interpretation. Finally, the hybrid specification captures the benefits of tuning hyperparameters at each step of the algorithm. The tuning parameters can be adjusted not only for prediction accuracy but also for possibly incorporating specific business objectives in the decision-making process. The transparency and interpretability of the models are advantages to convince interested parties, such as investors and regulators, that hybrid tree-based methods are much more than just a black box.

https://doi.org/10.1017/S0269964823000074 Published online by Cambridge University Press

We examine the prediction performance of our proposed models using datasets produced from a simulation study and a real-life insurance portfolio. We focus on comparison to the Tweedie GLM since this has become widespread for modeling claims arising from short-term insurance policies. Broadly speaking, hybrid-tree models can outperform the Tweedie GLM without loss of interpretability. However, there are opportunities to improve and to widen the scope of this hybrid structure that can be explored for further research. When y_{i_f} is a count rather than binary, y_{i_s} can be the total claim amount, the average claim amount, or even the vector of individual claim amounts. It depends on what kind of hybrid structure we choose to deploy. For example, we can use multivariate (or multi-response) regression models at the terminal nodes when we have detailed severity information regarding individual claim amounts and frequency information recording the number of claims. In addition, it is possible to improve hybrid tree-based methods by considering (1) models to accommodate the highly imbalanced nature of the claim frequency component; (2) additional hierarchical models for comparative purposes; (3) models to accommodate non-Gaussian structures for the claim severity component; and (4) other loss functions to optimize that justify the hybrid structure.

Funding statement. We thank the financial support of the Society of Actuaries through its Centers of Actuarial Excellence (CAE) grant for our research project on *Applying Data Mining Techniques in Actuarial Science*.

Conflicts of Interest. The authors declare no conflict of interest.

References

[1] Breiman, L., Friedman, J.H., Olshen, R.A., & Stone, C.J. (1984). *Classification and regression trees*. Boca Raton, FL: Taylor & Francis.
[2] Chaudhuri, P., Lo, W.-D., Loh, W.-Y., & Yang, C.-C. (1995). Generalized regression trees. *Statistica Sinica* 5(2): 641–666.
[3] De Mol, C., De Vito, E., & Rosasco, L. (2009). Elastic-net regularization in learning theory. *Journal of Complexity* 25(2): 201–230.
[4] Donoho, D.L. & Johnstone, I.M. (1995). Adapting to unknown smoothness via wavelet shrinkage. *Journal of the American Statistical Association* 90(432): 1200–1224.
[5] Frees, E.W., Derrig, R.A., & Meyers, G. (2014). *Predictive modeling applications in actuarial science: Volume I predictive modeling techniques*. New York, NY: Cambridge University Press.
[6] Frees, E.W., Lee, G., & Yang, L. (2016). Multivariate frequency-severity regression models in insurance. *Risks* 4(4): 1–36.
[7] Friedman, J., Hastie, T., Höfling, H., & Tibshirani, R. (2007). Pathwise coordinate optimization. *The Annals of Applied Statistics* 1(2): 302–332.
[8] Fu, W.J. (1998). Penalized regressions: The bridge versus the Lasso. *Journal of Computational and Graphical Statistics* 7(3): 397–416.
[9] Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction*. New York: Springer.
[10] Henckaerts, R., Côté, M.-P., Antonio, K., & Verbelen, R. (2021). Boosting insights in insurance tariff plans with tree-based machine learning methods. *North American Actuarial Journal* 25(2): 255–285.
[11] Hoerl, A.E. & Kennard, R.W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics* 12(1): 55–67.
[12] Hothorn, T., Hornik, K., & Zeileis, A. (2006). Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical Statistics* 15(3): 651–674.
[13] James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning*. New York, NY: Springer.
[14] Jørgensen, B. (1987). Exponential dispersion models. *Journal of the Royal Statistical Society: Series B (Methodological)* 49(2): 127–145.
[15] Klugman, S.A., Panjer, H.H., & Willmot, G.E. (2012). *Loss models: From data to decisions*. Hoboken, NJ: Wiley.
[16] Loh, W.-Y. (2006). Regression tree models for designed experiments. In J. Rojo (ed.), *Lecture Notes-Monograph Series*, vol. 49. Beachwood, OH: Institute of Mathematical Statistics, pp. 210–228.
[17] Loh, W.-Y. (2009). Improving the precision of classification trees. *The Annals of Applied Statistics* 3(4): 1710–1737.
[18] Nelder, J.A. & Wedderburn, R.W.M. (1972). Generalized linear models. *Journal of the Royal Statistical Society: Series A (General)* 135(3): 370–384.
[19] Parodi, P. (2012). Computational intelligence with applications to general insurance: A review: I – The role of statistical learning. *Annals of Actuarial Science* 6(2): 307–343.
[20] Quan, Z. & Valdez, E.A. (2018). Predictive analytics of insurance claims using multivariate decision trees. *Dependence Modeling* 6(1): 377–407.

[21] Quinlan, J.R. (1992b). Learning with continuous classes. In *5th Australian Joint Conference on Artificial Intelligence*, vol. 92. Singapore: World Scientific, pp. 343–348.
[22] Quinlan, J.R. (1992a). *C4.5: Programs for machine learning*. San Mateo, CA: Morgan Kaufmann.
[23] Smyth, G.K. & Jørgensen, B. (2002). Fitting Tweedie's compound Poisson model to insurance claims data: Dispersion modelling. *ASTIN Bulletin: The Journal of the International Actuarial Association* 32(1): 143–157.
[24] Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society: Series B (Methodological)* 58(1): 267–288.
[25] Xacur, O.A.Q. & Garrido, J. (2015). Generalised linear models for aggregate claims: To Tweedie or not?. *European Actuarial Journal* 5(1): 181–202.
[26] Zeileis, A., Hothorn, T., & Hornik, K. (2008). Model-based recursive partitioning. *Journal of Computational and Graphical Statistics* 17(2): 492–514.
[27] Zou, H. & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67(2): 301–320.

Appendix A. Further details for claim severity models

The following content provides details on the various forms of linear models for the claim severity.

After the construction of the claim frequency tree-based model, at each terminal node R_m, the linear coefficient $\boldsymbol{\beta}_{s_m} = (\beta_{s_m 0}, \beta_{s_m 1}, \ldots, \beta_{s_m p})^{\mathrm{T}}$ can be determined by

$$\widehat{\boldsymbol{\beta}}_{s_m} = \underset{\boldsymbol{\beta}_{\mathbf{sm}}}{\operatorname{argmin}} \frac{1}{n_m} \sum_{i=1}^{n_m} \ell\left(y_i, \beta_{s_m 0} + \sum_{j=1}^{p} x_{ij} \beta_{s_m j}\right), \tag{A.1}$$

where $\ell(y_i, \beta_{s_m 0} + \sum_{j=1}^{p} x_{ij}\beta_{s_m j})$ is the negative log-likelihood for sample i and n_m is the number of observations in the terminal node. For the Gaussian family, denoting the design matrix as $\mathbf{X}$, the estimator for the coefficient is well known as

$$\widehat{\boldsymbol{\beta}}_{s_m} = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1} \mathbf{X}^{\mathrm{T}}\mathbf{Y}.$$

Ridge regression [11] achieves better prediction accuracy compared to ordinary least squares because of the bias-variance trade-off. In other words, the reduction in the variance term of the coefficient is larger than the increase in its squared bias. It performs coefficient shrinkage and forces its correlated explanatory variables to have similar coefficients. In ridge regression, at each terminal node R_m, the linear coefficient $\boldsymbol{\beta}_{\mathbf{s_m}}$ can be determined by

$$\widehat{\boldsymbol{\beta}}_{s_m} = \underset{\boldsymbol{\beta}_{\mathbf{sm}}}{\operatorname{argmin}} \frac{1}{n_m} \sum_{i=1}^{n_m} \ell\left(y_i, \beta_{s_m 0} + \sum_{j=1}^{p} x_{ij} \beta_{s_m j}\right) + \lambda \sum_{j=1}^{p} \beta_{s_m j}^2, \tag{A.2}$$

where λ is a tuning hyperparameter that controls shrinkage, and thus the number of selected explanatory variables. For the following discussion, we assume that the values of x_{ij} are standardized so that

$$\frac{1}{n_m} \sum_{i=1}^{n_m} x_{ij} = 0 \quad \text{and} \quad \sum_{i=1}^{n_m} x_{ij}^2 = 1.$$

If the explanatory variables do not have the same scale, the shrinkage may not be fair. In the case of ridge regression within the Gaussian family, the coefficient in Eq. (A.2) can be shown to have the explicit form

$$\widehat{\boldsymbol{\beta}}_{s_m} = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X} + \lambda \mathbf{I}\right)^{-1} \mathbf{X}^{\mathrm{T}}\mathbf{Y},$$

where I is the identity matrix of appropriate dimension.

https://doi.org/10.1017/S0269964823000074 Published online by Cambridge University Press

Ridge regression does not automatically select the important explanatory variables. However, LASSO regression [24] has the effect of sparsity, which forces the coefficients of the least important explanatory variable to have a zero coefficient, therefore making the regression model more parsimonious. In addition, LASSO performs coefficient shrinkage and selects only one explanatory variable from the group of correlated explanatory variables. In LASSO, at each terminal node R_m, the linear coefficient $\boldsymbol{\beta}_{s_m}$ can be determined by

$$\widehat{\boldsymbol{\beta}}_{s_m} = \underset{\boldsymbol{\beta}_{\mathbf{sm}}}{\operatorname{argmin}} \frac{1}{n_m} \sum_{i=1}^{n_m} \ell\left(y_i, \beta_{s_m 0} + \sum_{j=1}^{p} x_{ij}\beta_{s_m j}\right) + \lambda \sum_{j=1}^{p} |\beta_{s_m j}|, \tag{A.3}$$

where λ is a tuning hyperparameter that controls shrinkage. For regularized least squares with LASSO penalty, Eq. (A.3) leads to the following quadratic programming problem

$$\widehat{\boldsymbol{\beta}}_{s_m} = \underset{\boldsymbol{\beta}_{s_m}}{\operatorname{argmin}} \frac{1}{2n_m} \sum_{i=1}^{n_m} \left(y_i - \beta_{s_m 0} - \sum_{j=1}^{p} x_{ij}\beta_{s_m j}\right)^2 + \lambda \sum_{j=1}^{p} |\beta_{s_m j}|.$$

Originally, [24] used the combined quadratic programming method to numerically solve for $\widehat{\boldsymbol{\beta}}_{s_m}$. In a later development, [8] proposed the shooting method and [7] redefined shooting as the coordinate descent algorithm, which is a popular algorithm used in optimization.

To illustrate the coordinate descent algorithm, we start with $\widehat{\boldsymbol{\beta}}_{s_m} = \mathbf{0}$. Given $\beta_{s_m j}$, $j \neq k$, the optimal $\beta_{s_m k}$ can be found by

$$\widehat{\beta}_{s_m k} = \underset{\beta_{s_m k}}{\operatorname{argmin}} \frac{1}{2n_m} \sum_{i=1}^{n_m} \left(y_i - \beta_{s_m 0} - \sum_{j \neq k} x_{ij}\beta_{s_m j} - x_{ik}\beta_{s_m k}\right)^2 + \lambda \sum_{j \neq k} |\beta_{s_m j}| + \lambda |\beta_{s_m k}|,$$

where $\lambda \sum_{j \neq k} |\beta_{s_m j}|$ is constant and can be dropped. We denote $\tilde{y}_i = y_i - \beta_{s_m 0} - \sum_{j \neq k} x_{ij}\beta_{s_m j}$. It follows that

$$\begin{aligned}
\widehat{\beta}_{s_m k} &= \underset{\beta_{s_m k}}{\operatorname{argmin}} \frac{1}{2n_m} \sum_{i=1}^{n_m} (\tilde{y}_i - x_{ik}\beta_{s_m k})^2 + \lambda |\beta_{s_m k}| \\
&= \underset{\beta_{s_m k}}{\operatorname{argmin}} \frac{1}{2n_m} \left(\sum_{i=1}^{n_m} \tilde{y}_i{}^2 + \beta_{s_m k}^2 \sum_{i=1}^{n_m} x_{ik}^2 - 2\beta_{s_m k} \sum_{i=1}^{n_m} \tilde{y}_i x_{ik} \right) + \lambda |\beta_{s_m k}| \\
&= \underset{\beta_{s_m k}}{\operatorname{argmin}} \frac{1}{2n_m} \left(\sum_{i=1}^{n_m} x_{ik}^2 \right) \left(\beta_{s_m k} - \frac{\sum_{i=1}^{n_m} \tilde{y}_i x_{ik}}{\sum_{i=1}^{n_m} x_{ik}^2} \right)^2 + \lambda |\beta_{s_m k}| + \text{constant} \\
&= \underset{\beta_{s_m k}}{\operatorname{argmin}} \left(\beta_{s_m k} - \frac{\sum_{i=1}^{n_m} \tilde{y}_i x_{ik}}{\sum_{i=1}^{n_m} x_{ik}^2} \right)^2 + \frac{\lambda}{\frac{1}{2n_m} \sum_{i=1}^{n_m} x_{ik}^2} |\beta_{s_m k}|.
\end{aligned}$$

This can be solved by the soft-thresholding Lemma discussed in [4].

Lemma A.1. (Soft-thresholding Lemma). *The following optimization problem*

$$\widehat{\beta}(t, \lambda) = \underset{\beta}{\operatorname{argmin}} (\beta - t)^2 + \lambda |\beta|$$

https://doi.org/10.1017/S0269964823000074 Published online by Cambridge University Press

has the solution of

$$\widehat{\beta}(t,\lambda) = (|t| - \lambda/2)^{+}\mathrm{sgn}(t)$$
$$= \begin{cases} t - \frac{\lambda}{2}, & \text{if } t > 0 \text{ and } \frac{\lambda}{2} < |t|, \\ t + \frac{\lambda}{2}, & \text{if } t < 0 \text{ and } \frac{\lambda}{2} < |t|, \\ 0, & \frac{\lambda}{2} \geq |t|. \end{cases}$$

Therefore, $\widehat{\beta}_{s_m k}$ can be expressed as

$$\widehat{\beta}_{s_m k} = \left(\frac{\sum_{i=1}^{n_m} |\tilde{y}_i x_{ik}|}{\sum_{i=1}^{n_m} x_{ik}^2} - \frac{\lambda}{\frac{1}{n_m}\sum_{i=1}^{n_m} x_{ik}^2} \right)^{+} \mathrm{sgn}\left(\sum_{i=1}^{n_m} \tilde{y}_i x_{ik} \right)$$

For $j = 1, \ldots, p$, update $\widehat{\beta}_{s_m k}$ by soft-thresholding when $\beta_{s_m j}, j \neq k$ takes the previously estimated value. Repeat loop until convergence.

Appendix B. Performance validation measures

The following table provides details of the various validation measures that were used throughout this paper to compare different predictive models.

Table A1. *Performance validation measures.*

Validation measures	Definition	Interpretation
Gini Index	$\text{Gini} = 1 - \frac{2}{N-1}\left(N - \frac{\sum_{i=1}^{N} i\tilde{y}_i}{\sum_{i=1}^{N} \tilde{y}_i}\right)$ where $\tilde{y}$ corresponds to y after ranking the corresponding predicted values $\widehat{y}$.	Higher Gini is better.
Coefficient of determination	$R^2 = 1 - \frac{\sum_{i=1}^{N} (\widehat{y}_i - y_i)^2}{\sum_{i=1}^{N} \left(y_i - \frac{1}{n}\sum_{i=1}^{n} y_i\right)^2}$ where $\widehat{y}$ is predicted values.	Higher R^2 is better.
Concordance correlation coefficient	$\text{CCC} = \frac{2\rho\sigma_{\widehat{y}_i}\sigma_{y_i}}{\sigma_{\widehat{y}_i}^2 + \sigma_{y_i}^2 + (\mu_{\widehat{y}_i} - \mu_{y_i})^2}$ where $\mu_{\widehat{y}_i}$ and μ_{y_i} are the means $\sigma_{\widehat{y}_i}^2$ and $\sigma_{y_i}^2$ are the variances ρ is the correlation coefficient	Higher CCC is better.
Root mean squared error	$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (\widehat{y}_i - y_i)^2}$	Lower RMSE is better
Mean absolute error	$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N} \lvert\widehat{y}_i - y_i\rvert$	Lower MAE is better.
Mean absolute percentage error	$\text{MAPE} = \frac{1}{N}\sum_{i=1}^{N} \left\lvert\frac{\widehat{y}_i - y_i}{y_i}\right\rvert$	Lower MAPE is better.
Mean percentage error	$\text{MPE} = \frac{1}{N}\sum_{i=1}^{N} \frac{\widehat{y}_i - y_i}{y_i}$	Lower |MPE| is better.

https://doi.org/10.1017/S0269964823000074 Published online by Cambridge University Press

Appendix C. Classification tree for the claim frequency

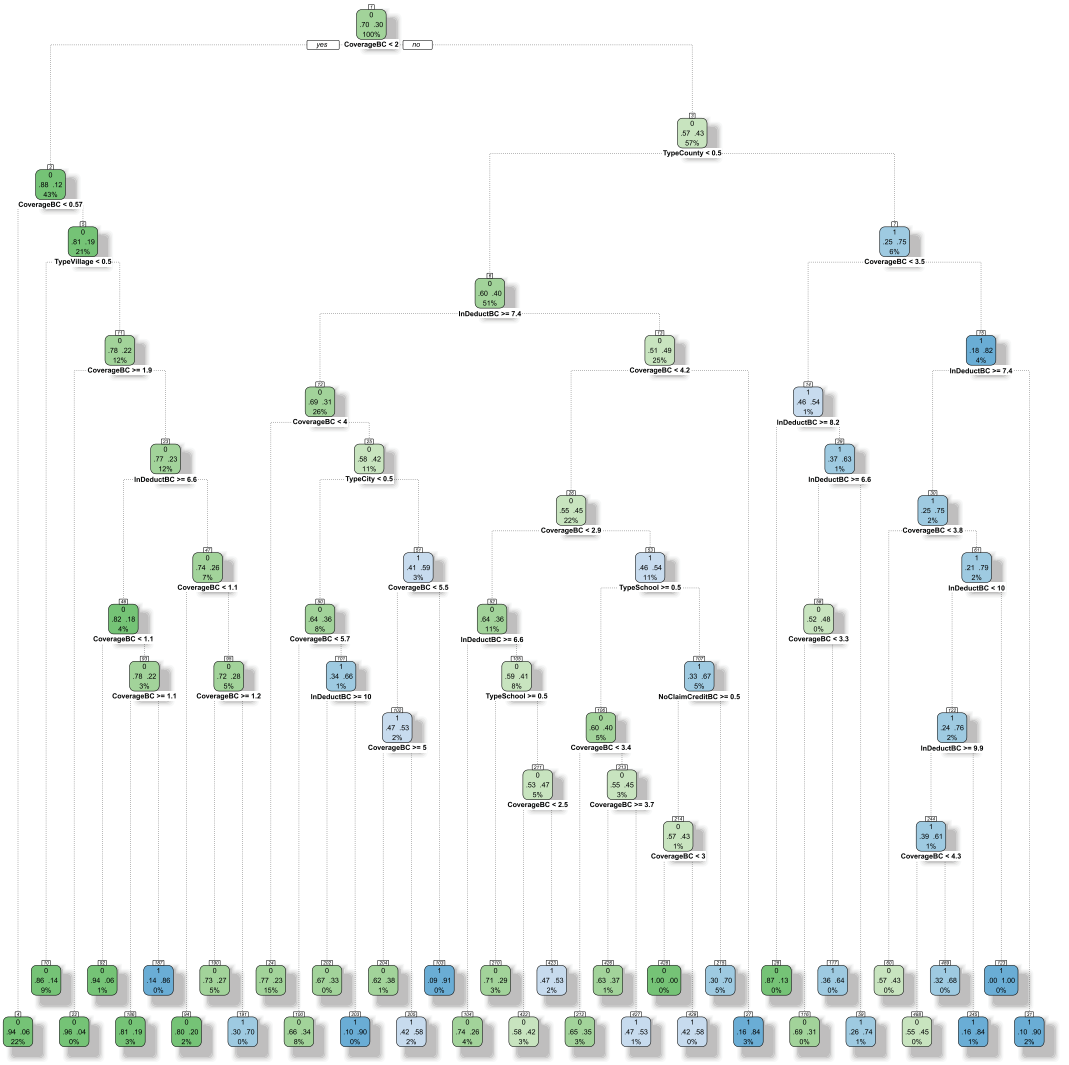

Figure A1. *Classification tree for the claim frequency.*

https://doi.org/10.1017/S0269964823000074 Published online by Cambridge University Press

Appendix D. R code for simulation study

```
GenerateData = function (nSample, nRealCatX, nFakeCatX,
                         nRealConX, nFakeConX, pho) {
  set.seed(777)

  n <- nSample
  p <- nRealCatX + nFakeCatX + nRealConX + nFakeConX

  p1 <- nRealConX + nFakeConX
  Cov <- outer(1:p1, 1:p1, function(x,y) {pho^abs(x-y)})
  xCon <- MASS::mvrnorm(n, rep(0, p1), Cov)

  p2 <- nRealCatX + nFakeCatX
  xCat <- NULL
  for(i in 1:p2){
    xCat <- cbind(xCat, sample(c(-3,-2,1,4), size = n, replace = T,
                               prob = c(0.25,0.25,0.25,0.25)))
  }

  muPoi <- exp(
     -0.1 +
      0.5 * apply(as.matrix(xCon[, 1:round(nRealConX/2)]),
                  1, sum) +
      0.1 * apply(as.matrix(xCon[, (round(nRealConX/2) + 1):nRealConX]),
                  1, sum) +
     -0.5 * apply(as.matrix(xCat[, 1:round(nRealCatX/2)]),
                  1, sum) +
      0.1 * apply(as.matrix(xCat[, (round(nRealCatX/2) + 1):nRealCatX]),
                  1, sum)
     )

  muTruePoi <- 1*muPoi/mean(muPoi)

  muGamma <- exp(
    6 +
    0.5 * apply(as.matrix(xCon[, 1:round(nRealConX/2)]),
                1, sum) +
   -0.1 * apply(as.matrix(xCon[, (round(nRealConX/2) + 1):nRealConX]),
                1, sum) +
    0.5 * apply(as.matrix(xCat[, 1:round(nRealCatX/2)]),
                1, sum) +
   -0.1 * apply(as.matrix(xCat[, (round(nRealCatX/2) + 1):nRealCatX]),
                1, sum)
   )

  muTrueGamma <- 10,000 * muGamma/mean(muGamma)

  power = 1.5
  phi = 2
  lambda <- muTruePoi^(2 - power)/(phi * (2 - power))
```

https://doi.org/10.1017/S0269964823000074 Published online by Cambridge University Press

```
  alpha <- (2 - power)/(1 - power)
  gam <- phi * (power - 1) * muTrueGamma^(power - 1)

  yPoi <- rpois(n, lambda = lambda)
  y <- array(dim = n, NA)

  for (i in (1:n)) {
    y[i] <- rgamma(1, shape = -yPoi[i] * alpha, scale = gam[i])
    if (y[i] > 0) {y[i] = y[i] * (1 + 0.25 * abs((rnorm(1))))}
  }

  dataTweedie = as.data.frame(cbind(xCat,xCon,y))
  dataTweedie[,1:p2] = lapply(dataTweedie[,1:p2], factor)

  return(dataTweedie)
}

SMdata <- GenerateData(10,000,20,10,20,10,0.5)
```

Cite this article: Quan Z., Wang Z., Gan G. and Valdez EA. (2023). On hybrid tree-based methods for short-term insurance claims. *Probability in the Engineering and Informational Sciences* **37**, 597–620. https://doi.org/10.1017/S0269964823000074