forecasting

MDPI

Article

Using Machine Deep Learning AI to Improve Forecasting of Tax Payments for Corporations

Charles Swenson

Marshall School of Business, University of Southern California, Los Angeles, CA 90089, USA; cswenson@marshall.usc.edu; Tel.: +1-213-740-4854

Abstract: This paper aims to demonstrate how machine deep learning techniques lead to relatively accurate forecasts of quarterly corporate income tax payments. Using quarterly data from Compustat for all U.S. publicly traded corporations from 2000 to 2024, I show that neural nets, the tree method, and random forest models provide robust forecasts despite their encompassing COVID-19 pandemic time periods. The results should be of interest to corporate tax planners, stock analysts, and governments.

Keywords: machine learning; taxation; forecasts

check for updates

Citation: Swenson, C. Using Machine Deep Learning AI to Improve Forecasting of Tax Payments for Corporations. *Forecasting* **2024**, *6*, 968–984. https://doi.org/10.3390/forecast6040048

Academic Editors: Konstantinos Nikolopoulos and Vasileios Bougioukos

Received: 23 September 2024
Revised: 18 October 2024
Accepted: 19 October 2024
Published: 25 October 2024

Copyright: © 2024 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

A vexing problem in corporate finance is the accurate prediction and pre-payment of corporate Federal income taxes. Any corporation that expects to have more than USD 500 in taxes due must pre-pay at least 90% of the predicted annual tax bill in quarterly installments or face a penalty of 2% above the federal funds rate, e.g., a penalty of 8% at current (2024) rates. In 2020, corporations paid over USD 199 million in such penalties (See IRS *Corporate Statistics of Income* available at SOI Tax Stats—Corporation Income Tax Returns Complete Report (Publication 16)—Internal Revenue Service (irs.gov) access on 23 September 2024). Companies devote resources to accurately forecast such payments to avoid underpayment penalties and overpayment when such funds could be invested elsewhere. Companies try to forecast such payments based on predicted income for the year, and often times, simply make payments equal to previous quarter (or even year) payments as a rule of thumb (Typically, companies attempt to predict total taxes due for the years, and divide this annual number by four to have equal quarterly payments. See Bratten et al. [1], who discuss management's use of (often erroneous) heuristics to forecast effective tax rates for their company. See Hirst et al. [2], who discuss high variations and inaccuracies in managers' company earnings forecasts (taxes are a significant portion of earnings). See also Blaufus et al. [3] for a literature review discussing widespread management misperceptions and forecasting of taxes). Similarly, stock analysts forecast corporate quarterly taxes in the process of evaluating a company's stock value (See Bratten et al. [1]). Here, better forecasts can improve such valuations and, thus, stock market efficiency. Such analyst and management forecasts of taxes were potentially made even more uncertain during the COVID-19 pandemic period.

A potential solution to this problem can be machine learning. Machine learning has emerged as a powerful tool for forecasting financial data, with applications ranging from stock price prediction to risk management. By utilizing algorithms that can learn patterns and relationships from historical data, machine learning models are able to make accurate predictions about future financial trends. One of the key advantages of machine learning in financial forecasting is its ability to handle large and complex datasets. Traditional statistical methods may struggle to capture the nuances and intricacies of financial data, especially when dealing with high-frequency trading or multiple variables. Machine

learning algorithms, on the other hand, can efficiently process vast amounts of data and identify patterns that human analysts may overlook.

Several machine learning techniques have been successfully applied to financial forecasting. One popular approach is time series analysis, which involves modeling the temporal dependencies in financial data. By using algorithms such as ARIMA (AutoRegressive Integrated Moving Average) or LSTM (Long Short-Term Memory), researchers have been able to predict stock prices and other financial metrics accurately. Another common technique is regression analysis, which involves fitting a model to the relationship between input variables and the target variable. By using algorithms like linear regression or support vector machines, researchers can forecast financial data based on historical trends and patterns.

In addition to these traditional machine learning methods, there has been a growing interest in using deep learning for financial forecasting. Deep learning algorithms, such as neural networks, have shown promise in capturing complex patterns in financial data and making accurate predictions. By training deep learning models on large datasets, researchers have achieved state-of-the-art results in stock price prediction and other financial tasks. See [2,4] for surveys of research and [5,6] for stock price prediction using ML. Despite the potential of machine learning in financial forecasting, challenges still need to be addressed. One of the main issues is data quality, as financial data can be noisy and prone to errors. Researchers need to carefully preprocess and clean the data before training machine learning models to ensure accurate predictions. Another challenge is model interpretability, as some machine learning algorithms can be black boxes that are difficult to interpret. Researchers need to develop techniques for explaining the predictions of their models, especially in high-stakes financial applications where transparency and accountability are crucial.

Overall, research on machine learning for financial forecasting is a rapidly growing field with exciting opportunities for innovation. By leveraging the power of algorithms and data, researchers can develop models that provide valuable insights and predictions for investors, traders, and financial institutions. As technology continues to advance, the potential for machine learning to revolutionize the financial industry is immense, and researchers are constantly exploring new techniques and methodologies to improve the accuracy and reliability of financial forecasts. More to the focus of this paper, and as noted in the next section, research on machine learning has shown promising results in improving tax compliance, fraud detection, and revenue forecasting. Machine learning algorithms have been utilized to analyze historical tax data, identify patterns, and make accurate predictions about future tax trends.

In this paper, I examine one such application—the ability of "deep learning" ML models (neural nets, random forest, and decision tree methods) to predict quarterly tax payments made by corporations. I use data from all U.S. publicly-traded companies (on Compustat) from 2000:Q1 through to 2024:Q1 (because the beginning time frame encompassed the pandemic and significant changes to corporate sales and tax payments, the models' robustness can be examined). I then compare the predictive powers of these models against simple models, which are the same as the last quarter's (SALQ) models often used by management. Despite the fact that COVID-19 pandemic-period data was used, the models used are quite robust. Since there is limited peer-reviewed research on forecasting corporate tax payments, this paper has the potential to significantly contribute to the literature and to practice. Moreover, the use of deep learning ML models is a novel approach compared to methods used in the literature on corporate taxation.

2. Deep Learning ML Research in Taxation

One key advantage of using machine learning in tax forecasting is its ability to handle large and complex datasets, which traditional methods may struggle with. Algorithms such as regression analysis, decision trees, and neural networks have been successfully applied to predict tax revenues, identify tax evasion, and optimize tax collection strategies.

For a general discussion, see [7,8]; Ref. [9] for tax payment default predictions; Ref. [10] for prediction of government revenues; Ref. [11] for tax crime prediction; Ref. [12] for optimizing tax administration policies; Ref. [13] for improving tax audit efficiency; Ref. [14] for using ML to predict outcomes in tax law; and [15] for the use of machine learning and advanced analytics in tax fraud detection. De Roux et al. [16] demonstrate the detection of potential fraudulent taxpayers using only unsupervised learning techniques and allowing the future use of supervised learning techniques. They demonstrate the ability of our model to identify taxpayers who are under-reporting on real tax payment declarations, reducing the number of potential fraudulent taxpayers to audit. Battaglini et al. [17] study how machine learning techniques can improve tax auditing efficiency using administrative data without the need for randomized audits. Using Italy's population data on sole proprietorship tax returns and audits, they find that replacing the 10% least promising audits with an equal number selected by their algorithm raises detected tax evasion by as much as 39%, and evasion that is actually paid back by 29%. Wahab et al. [18] propose descriptive and predictive analytics models for predicting Malaysia's digital economic income tax compliance. They show that the ensemble method can improve the single classification model's accuracy with the highest classification accuracy of 87.94% compared to the best single classification model.

Zumaya et al. [19] use a dataset of ten thousand tax evaders and build monthly and yearly temporal networks where nodes are contributors and directed links are invoices produced in a given time slice. Using deep neural networks and random forests, they achieve 90% accuracy and each method classifies more than 100 thousand suspects of tax evasion, with more than 40 thousand suspects classified by both methods. Murorunkwere et al. [20] evaluate various supervised machine learning models and find that the Artificial Neural Network method was the most robust model for predicting tax fraud in Rwanda for value-added taxes (VAT). Lee [21] uses neural nets to predict property tax valuation in Korea. Alsadhan [22] uses machine learning to analyze tax evasion in Saudi Arabia. Rahman et al. [23] use machine learning models based on five selected algorithms tested on the real dataset of 3365 Malaysian companies listed on Bursa Malaysia from 2005 to 2015. Liu et al. [24] use various machine learning methods and find that CEOs' hobby of flying airplanes is significantly associated with increased corporate tax avoidance. Kumar et al. [25] survey the literature and suggest that machines can help predict tax default (i.e., taxpayers unable to pay their taxes). Rabasco et al. [26] find that taxpayers identified by their machine learning models will, on average, increase their declared income by EUR 14,461—the average among all audited taxpayers being EUR 205. Lahan et al. [27] show that machine-learning techniques can be utilized to minimize risks and increase VAT compliance. Using a supervised learning classifier to predict tax subjects and corresponding tax rates based on related voucher information of journal reports, the methods reveal potential anomalies and compliance issues.

Visitpanya et al. [28] use machine learning to identify suspicious taxpayer behavior based on criteria derived from a literature review. Using public data from Thailand, their study yielded a precision of over 97% using selected supervised models. Using actual Italian taxpayer data, Didimio et al. [29] built four forecasting models for detecting the fiscal risk associated with each taxpayer. These models were logistic regression (LR), support vector machines (SVM), multi-layer perceptron neural networks (MLP), and random forests (RF). Masrom et al. [30] examined the performances of machine learning classification models for the detection of tax avoidance problems, finding that the support vector machine (SVM) method performed best. Uddin et al. [31] examined the application of machine learning for classifying the tax compliance level of manufacturing companies in Indonesia, finding that such models had 97% predictive power. Chen and Liu [32] collected Weibo posts from different users related to China's Annual Individual Income Tax Return (AIITR) and compared the performance of six machine learning models (support vector machine, random forest, naive Bayes, K-nearest neighbor, decision tree, and logistic regression), as well as a deep learning LSTM model, to examine Chinese taxpayer sentiment toward

the tax system. Mi et al. [33] propose and test a novel tax evasion detection method based on positive unlabeled (PU) learning with network embedding features (which they name "PUNE") using Chinese data. In a very unique study, Ebrahimzadeh et al. [34] examine neural pathway activity in subjects' brains to detect tax evasion behavior (See also Nener et al. [8] for a theoretical discussion of ML methods relating to taxation and wealth redistribution).

Although deep learning ML has provided promising results in a number of tax settings, as noted in the Introduction, there is very little research on the use of such methods in forecasting tax payments by corporations (Although not related to tax payment forecasting, corporate tax risk has been proposed as an area for the use of ML (see Liu et al. [24])). A relevant study is by Gunther et al. [35], who compare their machine-generated annual effective tax rate (ETR) predictions with those from ETRs implied by analysts' earnings forecasts and find the algorithm's predictions are less biased, more precise, and explain more of the variance in future ETRs. Like this study, they use Compustat data on publicly traded companies. However, they use *annual* data (this study uses *quarterly* data), and their focus is on ETRs, which are income tax expenses divided by pretax financial earnings. Both such measures are based on financial reporting, and the numerator is generally only loosely correlated with actual cash tax outlays paid to the government. Other related studies used various forecasting techniques to examine a single enterprise. Molapo et al. [36] modeled and predicted the tax revenue of an enterprise in South Africa using BVAR, ARIMA, and State Space exponential smoothing, thus providing help for the enterprise's budget and strategic planning. Rubin et al. [37] used a microsimulation model to forecast the amount of corporate tax revenue, using data from a sample of actual tax returns and making inferences about groups of individual or corporate taxpayers. In the first (and, to my knowledge, only) study to use deep learning in this area, Hui et al. [38] model a single Chinese firm and find that a hybrid random forest model fits the data well. This study goes beyond these studies by examining all (as opposed to a single firm) firms in the U.S. Further, this study shows that different ML models might be required across different firms.

3. Deep Learning ML Models

3.1. Neural Network Models

A neural network is a computational model inspired by the way the human brain processes information. It is composed of layers of interconnected nodes, called neurons, that work together to learn and make predictions from data. The *input layer* receives the initial data or features of the information being processed. Each neuron in the input layer represents a different feature of the data. *Hidden layers:* Between the input and output layers, there can be one or more hidden layers in a neural network. Each hidden layer consists of multiple neurons that perform computations on the input data using weights and biases. *Neurons:* Neurons in a neural network receive input signals, apply weights to them, sum them up, and then pass the result through an activation function. The activation function introduces non-linearity into the model, allowing the neural network to capture complex patterns in the data. *Weights and biases*: The connections between neurons are represented by weights, which are adjusted during training to minimize the error in the model's predictions. Biases are additional parameters that help shift the output of a neuron. *Output layer:* The output layer produces the final prediction or output of the neural network model. The number of neurons in the output layer depends on the type of problem being solved (e.g., regression, classification).

During training, the neural network learns from labeled data by adjusting the weights and biases to minimize the difference between the predicted and actual output. This is typically done using optimization algorithms like gradient descent. Once the neural network is trained, it can be used to make predictions on new, unseen data by passing the input through the network and obtaining the output from the output layer.

Neural networks are widely used in various fields, including image and speech recognition, natural language processing, and financial forecasting, due to their ability to

learn complex patterns from data. They are a fundamental component of deep learning and artificial intelligence systems.

3.2. Decision Tree Methods

A tree-based machine learning method, such as decision trees, random forests, or gradient boosting, is a type of algorithm that makes predictions by recursively partitioning the input space into a set of regions and assigning a prediction value to each region. A decision tree is a flowchart-like structure where each internal node represents a feature or attribute, each branch represents a decision rule, and each leaf node represents the outcome or prediction. Decision trees are built by recursively splitting the data based on the values of the features to create regions that are as homogenous as possible in terms of the target variable.

Tree-based methods can provide insights into the importance of different features in making predictions. Features frequently used near the tree's root or lead to significant decreases in impurity are considered more important in the prediction process. One of the advantages of tree-based machine learning methods is their interpretability. Decision trees provide a clear and intuitive representation of how the model makes predictions, making it easier to understand and explain the decision-making process.

Due to their simplicity, interpretability, and strong predictive performance, tree-based machine learning methods are widely used in various applications, including classification, regression, and feature selection tasks.

3.3. Random Forest Models

A random forest model is a popular ensemble learning technique that combines multiple decision trees to make more accurate predictions. This method creates decision trees (see previous discussion). The random forest method next uses ensemble learning, consisting of multiple decision trees, where each tree is trained on a random subset of the training data and a random subset of features. This randomness helps to reduce overfitting and improve the generalization of the model. The next method employs bagging, which is the process of training each decision tree on a random subset of the training data. This process is known as bagging (bootstrap aggregating). This helps to introduce diversity among the trees and reduce the variance of the model.

The method also employs random feature selection. In addition to training each tree on a random subset of the data, random forest also selects a random subset of features for each tree. This further enhances the diversity of the trees and prevents the model from being biased towards certain features. Finally, when making predictions, each decision tree in the random forest model independently predicts the outcome based on the input features. The final prediction is then determined by aggregating the predictions of all the trees, either through a voting mechanism (classification) or averaging (regression).

Random forest models are known for their high accuracy, robustness, and ability to handle large datasets with high-dimensional feature spaces. They are commonly used in various applications, including classification, regression, and feature selection tasks (Although not used here, gradient boosting is another ensemble learning method that builds a series of decision trees sequentially, where each subsequent tree is trained to correct the errors of the previous tree. The final prediction is the sum of the predictions of all the trees in the sequence).

4. Data

Quarterly financial statement data for all U.S. publicly traded firms from 2000 through the first quarter of 2024 is drawn from Compustat. Variables used are contained in Table 1 (I utilize over 30 of the most potentially useful (in predicting tax payments) data items given in the Compustat database). Since Compustat does not provide quarterly cash tax payments, we can derive them as taxes payable at the end of the previous quarter, income tax expense from the current quarter, and taxes payable at the end of the current quarter. In

this paper's machine learning analysis, I use all of the above variables (except txpq and txtq, which are used to derive the dependent variable cash taxes paid during the quarter). I employ three deep ML methods: neural nets, decision tree, and random forest. Each has its own advantages, but the study is agnostic in predicting which method will work best with these data.

Table 1. Data used in models.

Variable Name	Definition
fyearq	Fiscal Year (fyearq)
fqtr	Fiscal Quarter (fqtr)
conm	Company Name (conm)
curcdq	Currency
datacqtr	Calendar Data Year and Quarter (datacqtr)
datafqtr	Fiscal Data Year and Quarter (datafqtr)
atq	Assets—Total (atq)
ceqq	Common/Ordinary Equity—Total (ceqq)
cheq	Cash and Short-Term Investments (cheq)
chq	Cash (chq)
cogsq	Cost of Goods Sold (cogsq)
dlcq	Debt in Current Liabilities (dlcq)
dlttq	Long-Term Debt—Total (dlttq)
drcq	Deferred Revenue—Current (drcq)
drltq	Deferred Revenue—Long-term (drltq)
esubq	Equity in Earnings (I/S)—Unconsolidated Subsidiaries (esubq)
ibq	Income Before Extraordinary Items (ibq)
intanq	Intangible Assets—Total (intanq)
oiadpq	Operating Income After Depreciation—Quarterly (oiadpq)
oibdpq	Operating Income Before Depreciation—Quarterly (oibdpq)
piq	Pretax Income (piq)
ppegtq	Property, Plant and Equipment—Total (Gross)—Quarterly (ppegtq)
ppentq	Property Plant and Equipment—Total (Net) (ppentq)
revtq	Revenue—Total (revtq)
saleq	Sales/Turnover (Net) (saleq)
txdbaq	Deferred Tax Asset—Long Term (txdbaq)
txdbcaq	Current Deferred Tax Asset (txdbcaq)
txdbclq	Current Deferred Tax Liability (txdbclq)
txdbq	Deferred Taxes—Balance Sheet (txdbq)
txdiq	Income Taxes—Deferred (txdiq)
txditcq	Deferred Taxes and Investment Tax Credit (txditcq)
txpq	Income Taxes Payable (txpq)
txtq	Income Taxes—Total (txtq)
txwq	Excise Taxes (txwq)
xrdq	Research and Development Expense (xrdq)
xsgaq	Selling, General and Administrative Expenses (xsgaq)

Table 1. *Cont.*

Variable Name	Definition
cshtrq	Common Shares Traded—Quarter (cshtrq)
mkvaltq	Market Value—Total (mkvaltq)
prccq	Price Close—Quarter (prccq)
prchq	Price High—Quarter (prchq)
prclq	Price Low—Quarter (prclq)
incorp	Current State/Province of Incorporation Code (incorp)
loc	Current ISO Country Code—Headquarters (loc)
naics	North American Industry Classification Code (naics)
sic	Standard Industry Classification Code (sic)
state	State/Province (state)

5. Machine Learning Results

5.1. Aggregate Results

Although our objective is to use ML to provide forecasted cash tax payments for individual companies, it is instructive to examine if deep learning models can provide useful estimates at the aggregate level (using all companies on Compustat). There were 648,499 firm years of observations used. Seventy percent of the observations were used for training and 30 percent for validation (Computations were done using SAS Enterprise Miner's multithreaded, high-performance (HP) nodes). Table 2 shows model fits (squared errors) for decision trees, neural nets, and random forest models (although not shown above, regression (using generalized least squares) shows a model fit with less explanatory power/error than the above models). The decision tree method significantly outperformed the other models, with less than half of the validation errors for neural nets and random forest (*t*-tests showed that the mean squared errors were significantly different).

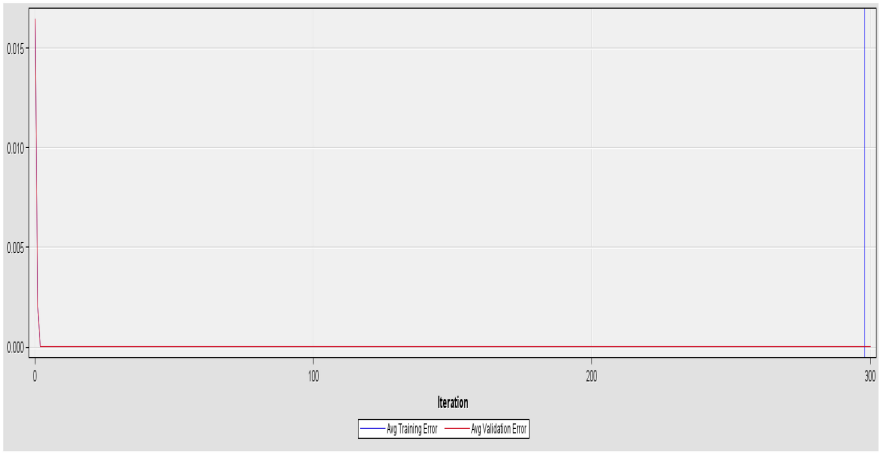
Table 2. All firms: model fits (squared errors) *.

	Training: Avg. Sq. Error	**Training: Sum of Sq. Errors**	**Validation: Avg. Sq. Error**	**Validation: Sum of Sq. Errors**
Decision Tree	36,448.36	1.65×10^{10}	91,904.77	1.79×10^{10}
Neural Nets	126,380.98	5.74×10^{10}	208,248.50	4.05×10^{10}
Random Forest	94,603.81	4.29×10^{10}	232,056.76	4.51×10^{10}

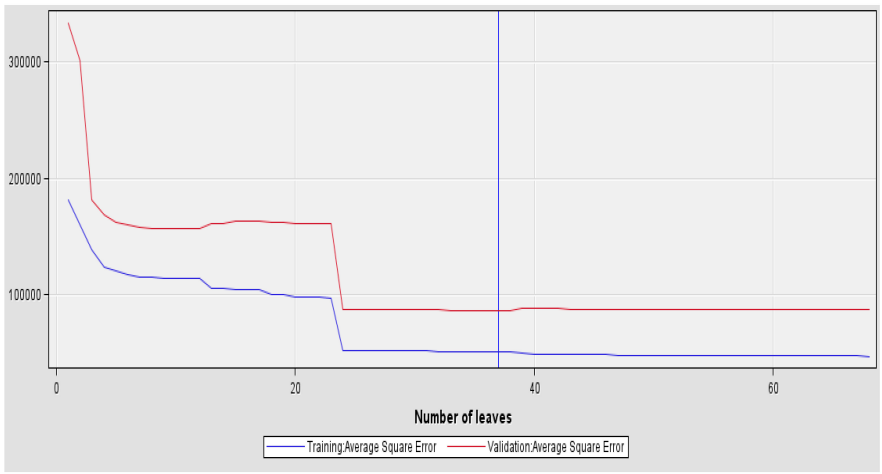
* amounts in USD thousands.

To examine model fits over various levels, the graphs in Figure 1A–C show mean predicted versus actual quarterly cash tax payments at various levels for the decision tree (the decision tree method used in this paper uses the CART (Classification and Regression Trees) algorithm, which is a powerful and widely-used method for building decision tree models. The CART algorithm works by recursively splitting the data into subsets based on the values of input variables, with the goal of creating homogeneous subgroups that are as pure as possible in terms of the target variable. The algorithm selects the best split at each node based on criteria such as Gini impurity or entropy for classification tasks and squared error or mean absolute error for regression tasks. The default splitting criterion is set here to Gini impurity or entropy, which are measures used to evaluate the quality of a split in a decision tree. These criteria help determine how to partition the data at each node to maximize information gain. The maximum tree depth is set to 20 here; this specifies the maximum number of levels or splits that the decision tree can have. This helps to control the complexity of the tree and prevent overfitting. The minimum parent node size is set to the default in the software used; the minimum parent node size specifies the minimum number

of observations required in a node to consider splitting it further. This helps to prevent the creation of nodes with too few observations, which may lead to overfitting. Missing values are estimated using imputation. Pruning is set to automatic pruning, where the decision tree is pruned after construction to improve generalization and reduce overfitting. Evaluation metrics used here are accuracy, misclassification rate, and Gini index), neural nets (to avoid overfitting (a potential problem with the modest data size here), input, hidden, and output layers are set to one for all neural net analyses in this paper), and random forest methods (here (and for all analyses in this paper) the CART (Classification and Regression Trees) algorithm is used. This algorithm works by recursively splitting the data into subsets based on the values of input variables, with the goal of creating homogeneous subgroups that are as pure as possible in terms of the target variable. The algorithm selects the best split at each node based on criteria such as Gini impurity or entropy for classification tasks and squared error or mean absolute error for regression tasks. Here, the number of trees is set to 100, and the number of input variables to consider at each split in the decision trees is set to the square root of the total number of input variables (this helps to introduce randomness and reduce overfitting in the model). The random seed (to start the randomization process) is created automatically by the software, and the overall evaluation metrics include measures such as accuracy, misclassification rate, and Gini index), for both training and validation results. Here again, the results show that the decision tree method is the best model fit in terms of error diagnostics.

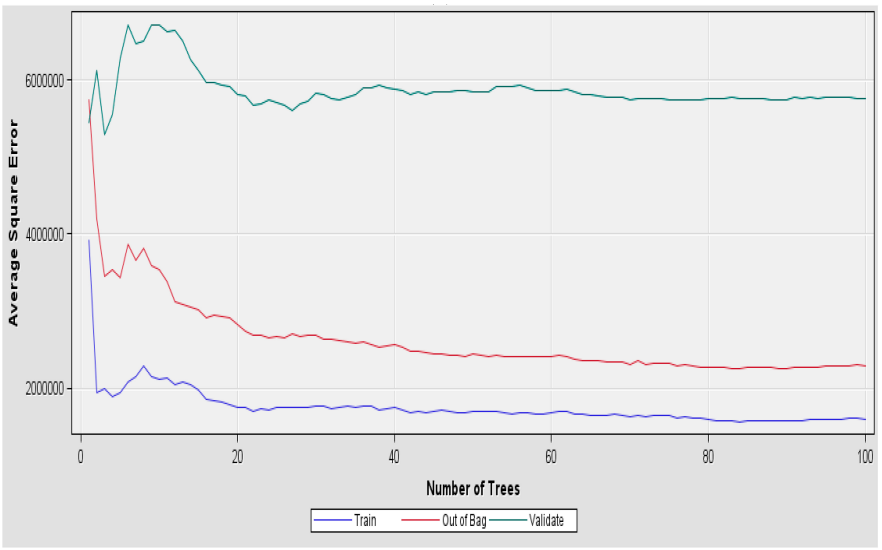


(**A**)



(**B**)

Figure 1. *Cont.*



(**C**)

Figure 1. All firms: (**A**) neural net model fits over iterations; (**B**) decision tree model fits over iterations; (**C**) random forest model fits over iterations.

5.2. Results for Individual Companies

There are over 3000 companies on the Compustat quarterly files, making applying ML to each of them impractical (given the scope of this paper). Accordingly, I examine three companies: Best Buy, Exxon Mobil, and Amazon.

5.2.1. Best Buy

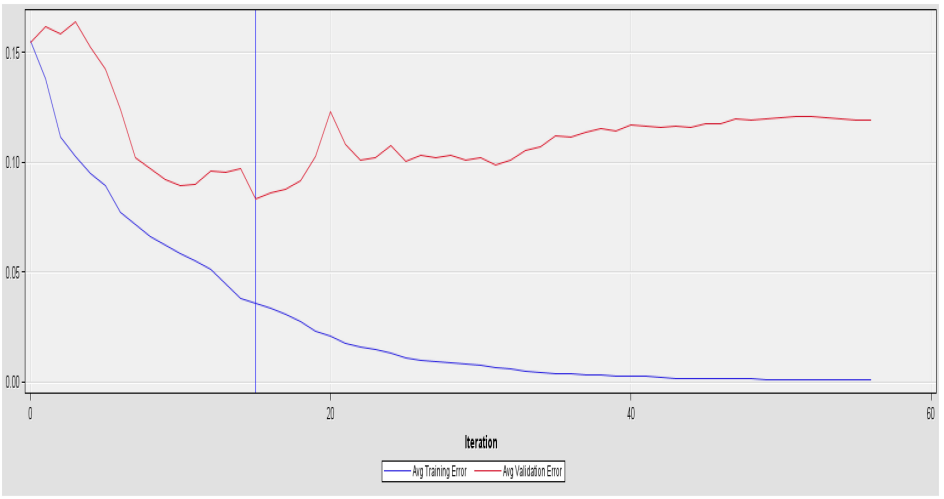
Best Buy is a national electronics retailer selling both in-store and online. Seventy percent of the observations were used for training and 30 percent for validation. Table 3 shows model fits (squared errors) for the neural nets, random forest, and decision tree models, respectively (although not shown above, regression results (using generalized least squares) show a model fit with less explanatory power/error than the above models). The graphs in Figure 2A–C show the mean predicted versus actual quarterly cash tax payments for the decision tree, neural nets, and random forest methods for training and validation results.

Table 3. Best Buy: model fits (squared errors) *.

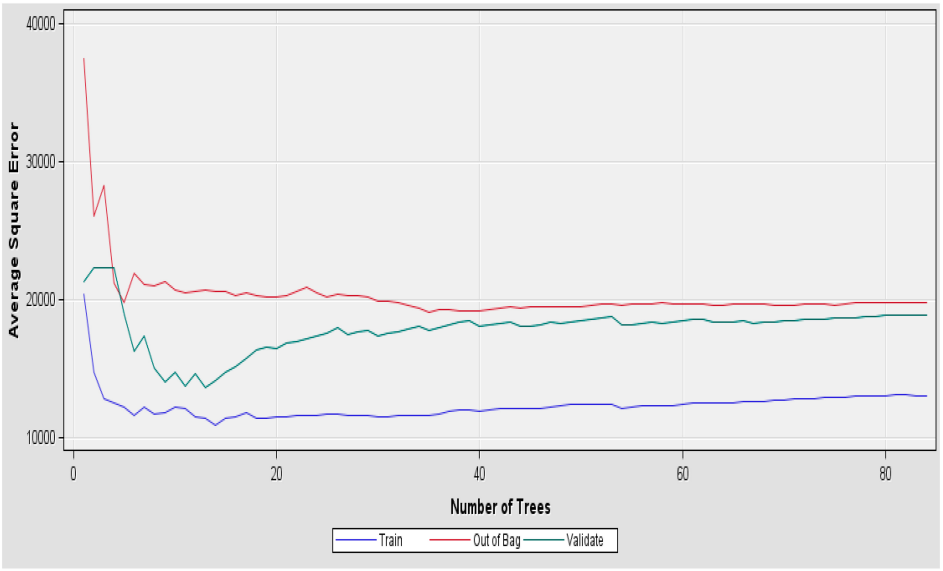
Model	Validation: Avg. Sq. Error	Training: Avg. Sq. Error	Training: Sum of Sq. Errors	Validation: Avg. Sq. Error
Neural Nets	11,900.76	9073.28	471,810.95	11,900.76
Random Forest	17,257.94	11,691.21	607,942.94	17,257.94
Decision Tree	18,958.88	13,859.49	720,693.50	18,958.88

* amounts in USD thousands.

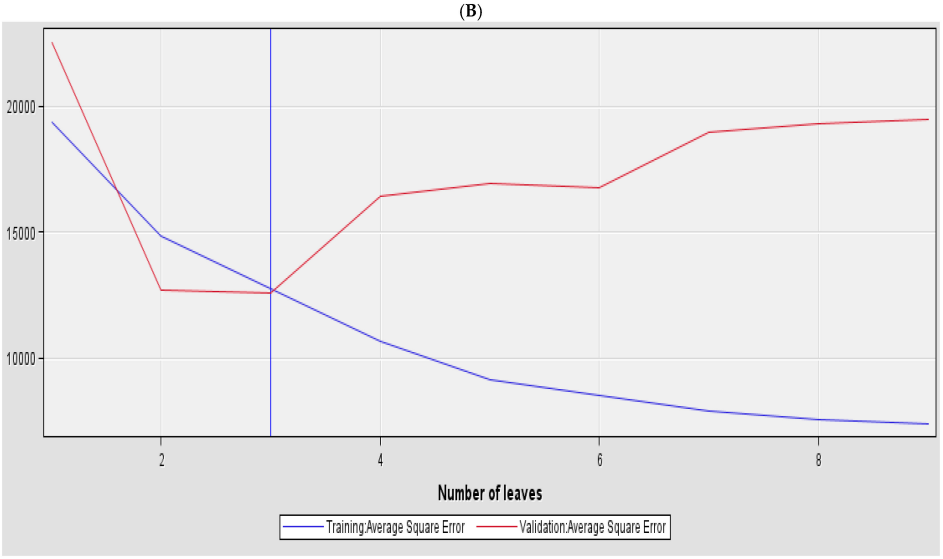
Overall, the neural nets model shows the lowest error in both the training and validation stages (*t*-tests showed that the mean squared errors were significantly different). To test the neural net model, we compare predicted values to actual ones and compare predicted values to a simple method typically employed by companies. This method assumes that the current quarter's payments are the same as the previous quarter's payments or "same as last quarter" (SALQ). Table 4 shows the percent error in predicted value versus actual. All observations (i.e., observations used in both training and validation) are used. As can be seen, the average prediction error for the neural net model is −0.008 percent, compared to the SALQ error of 1.0476 percent. Moreover, maximum and standard deviation errors are significantly higher for the SALQ method. Figure 2D shows actual versus predicted quarterly tax payments over time. As can be seen, the two-track well over time.



(**A**)

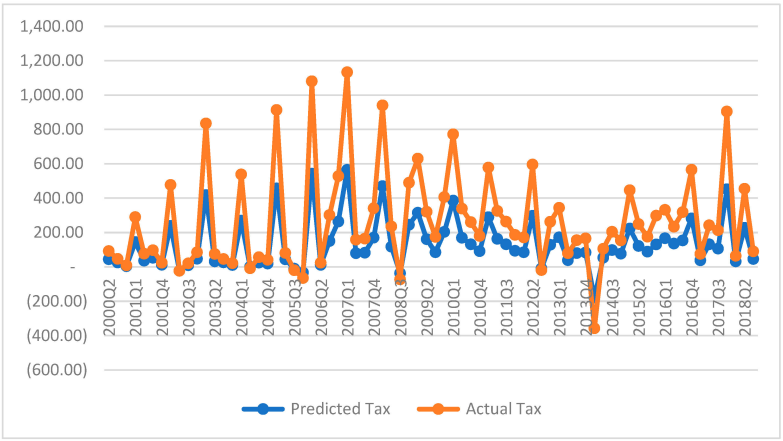


(**B**)



(**C**)

Figure 2. *Cont.*



(**D**)

Figure 2. Best buy: (**A**) neural nets model fits (actual versus predicted) over iterations; (**B**) random forest model fits (actual versus predicted) over iterations; (**C**) decision tree model fits (actual versus predicted) over iterations; (**D**) neural nets quarterly tax payments-actual versus predicted-over Time (USD thousands).

Table 4. Best Buy: prediction error percent *.

	Neural Nets	**SALQ**
Minimum	−1.0896	−17.8209
Maximum	0.3838	29.1871
Mean	−0.0088	1.0476
Standard deviation	0.1673	6.596

* amounts in USD thousands.

5.2.2. ExxonMobil

For Exxon Mobil, 70 percent of the observations were used for training and 30 percent for validation. Table 5 shows model fits (squared errors) for the neural nets, random forest, and decision tree models, respectively (*t*-tests showed that the mean squared errors were significantly different. Although not shown above, regression results (using generalized least squares) show a model fit with less explanatory power/error than the above models). The graphs in Figure 3A–C show mean predicted versus actual quarterly cash tax payments at various levels for the decision tree, neural nets, and random forest methods for both training and validation results to examine model fit evolution. Notably, the decision tree method performed poorly, resorting to overall means for each predicted value.

Table 5. Exxon Mobil: model fits (squared errors) *.

	Training: Avg. Sq. Error	**Training: Sum of Sq. Errors**	**Validation: Avg. Sq. Error**	**Validation: Sum of Sq. Errors**
Neural Nets	2,311,251.85	1.53×10^8	3,656,319.04	1.0×10^8
Decision Tree	1,157,324.52	7.64×10^7	4,353,659.39	1.2×10^8
Random Forest	1,520,682.85	1.00×10^8	5,498,649.23	1.5×10^8

* amounts in USD thousands.

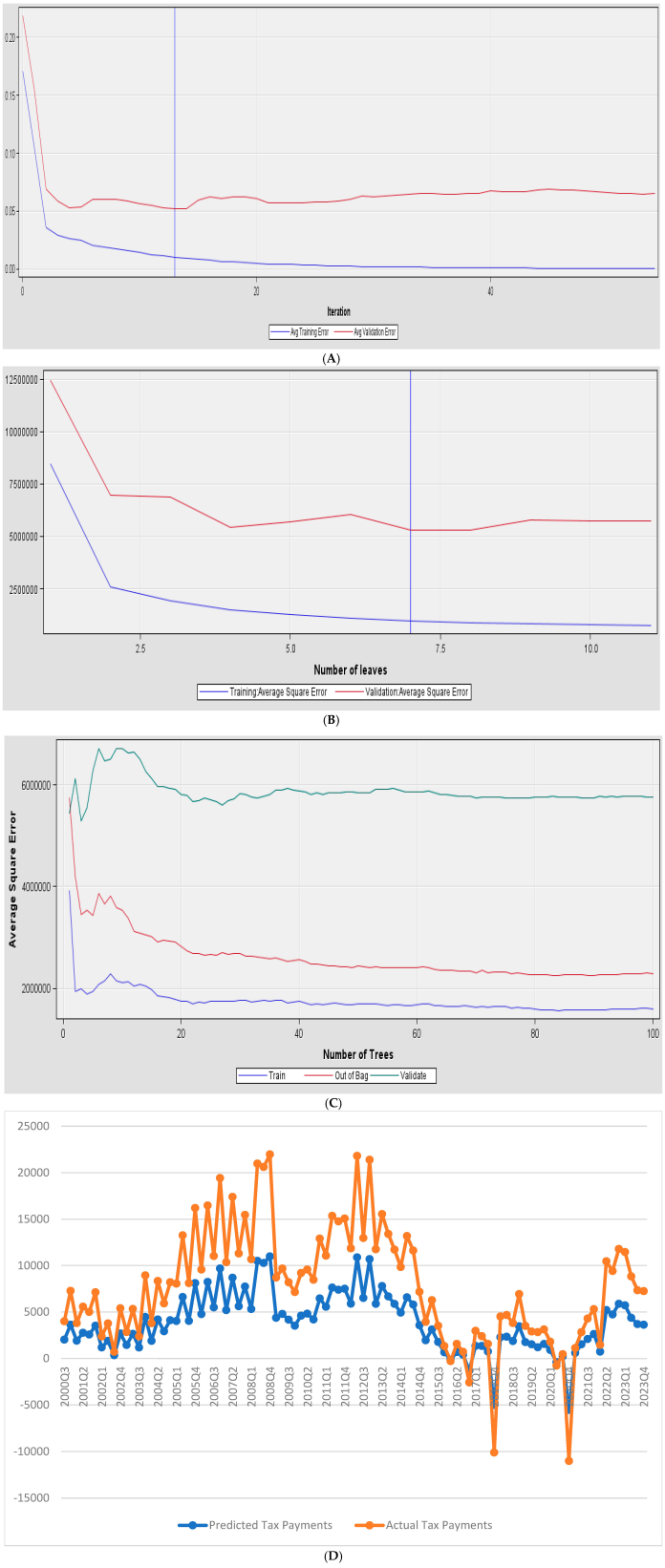


Figure 3. Exxon Mobil: (**A**) neural net model fits over iterations; (**B**) decision tree model fits over iterations; (**C**) random forest model fits over iterations; (**D**) neural nets quarterly tax payments-actual versus predicted-over time (USD thousands).

The neural net model shows the lowest error in both the training and validation stages (the neural net results are especially encouraging since this method is typically thought to be most effective with very large datasets (e.g., in the millions)). To test the neural nets model, we compare predicted values to actual and also compare predicted values to the simple model typically used by companies—assuming that the current quarter payments are the same as the previous quarter payments or the same as last quarter (SALQ). Table 6 shows the percent error in predicted value versus actual. All observations (i.e., observations used in both training and validation) are used. As can be seen, the average prediction error for the neural net model is −0.0047 percent, compared to the SALQ error of −0.38 percent. Moreover, maximum and standard deviation errors are significantly higher for the SALQ method. Figure 3D shows actual versus predicted quarterly tax payments over time. As can be seen, the two-track well over time.

Table 6. Exxon Mobil: prediction error percent *.

	Neural Nets	**SALQ**
Minimum	−0.2542	−25.6728
Maximum	0.2897	6.5167
Mean	0.0047	−0.3844
Standard deviation	0.0659	3.1276

* amounts in USD thousands.

5.2.3. Amazon

For Amazon, 70 percent of the observations were used for training and 30 percent for validation. Table 7 shows model fits (squared errors) for the neural nets, random forest, and decision tree models, respectively (*t*-tests showed that the mean squared errors were significantly different. Although not shown above, regression results (using generalized least squares) show a model fit with less explanatory power/error than the above models). The decision tree model shows the lowest error in both the training and validation stages. To test the decision tree model, we compare predicted values to actual and compare predicted values to a simple model, assuming that the current quarter's payments are the same as the previous quarter's payments or the same as the last quarter (SALQ). Table 8 shows the percent error in predicted value versus actual. All observations (i.e., observations used in both training and validation) are used. The average prediction error for the decision tree model is 0.1394 percent, compared to the SALQ error of −1.2729 percent. Moreover, maximum and standard deviation errors are significantly higher for the SALQ method.

Table 7. Amazon: model fits (squared errors) *.

	Training: Avg. Sq. Error	**Training: Sum of Sq. Errors**	**Validation: Avg. Sq. Error**	**Validation: Sum of Sq. Errors**
Decision Tree	90,402.66	5,966,575.27	25,787.05	722,037.46
Random Forest	92,162.51	6,082,725.64	38,438.56	1,076,279.54
Neural Net	214,067.70	14,128,468.15	88,871.72	2,488,408.24

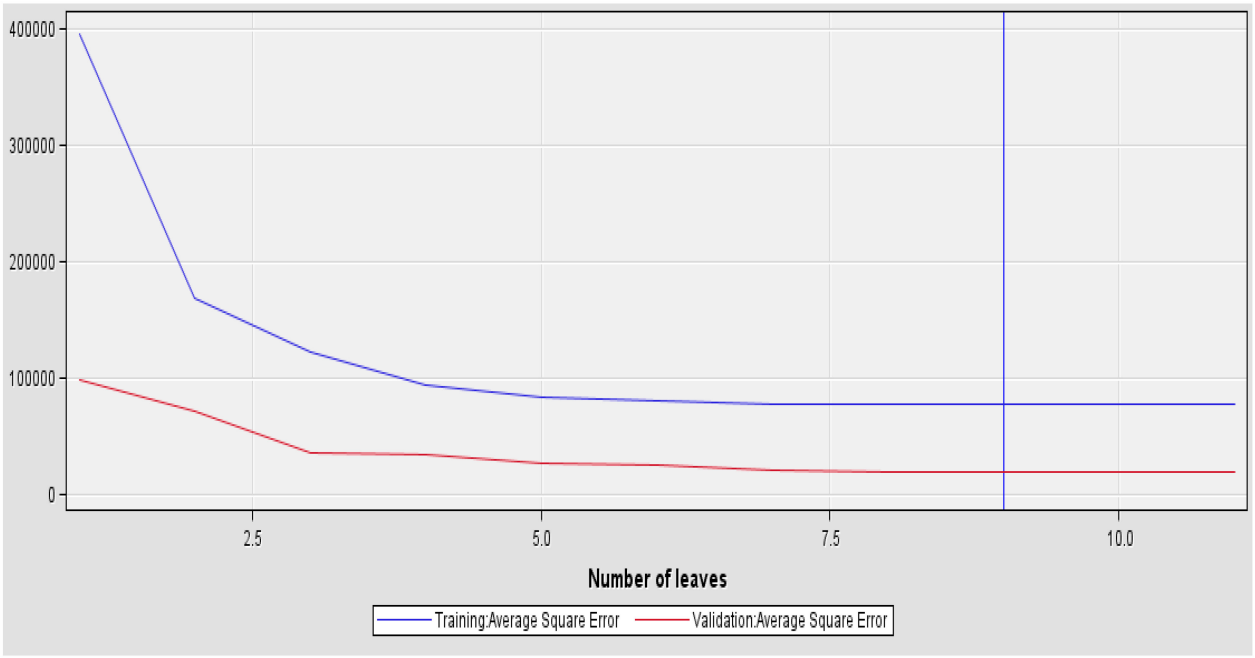
* amounts in USD thousands.

Table 8. Amazon: prediction errors percent *.

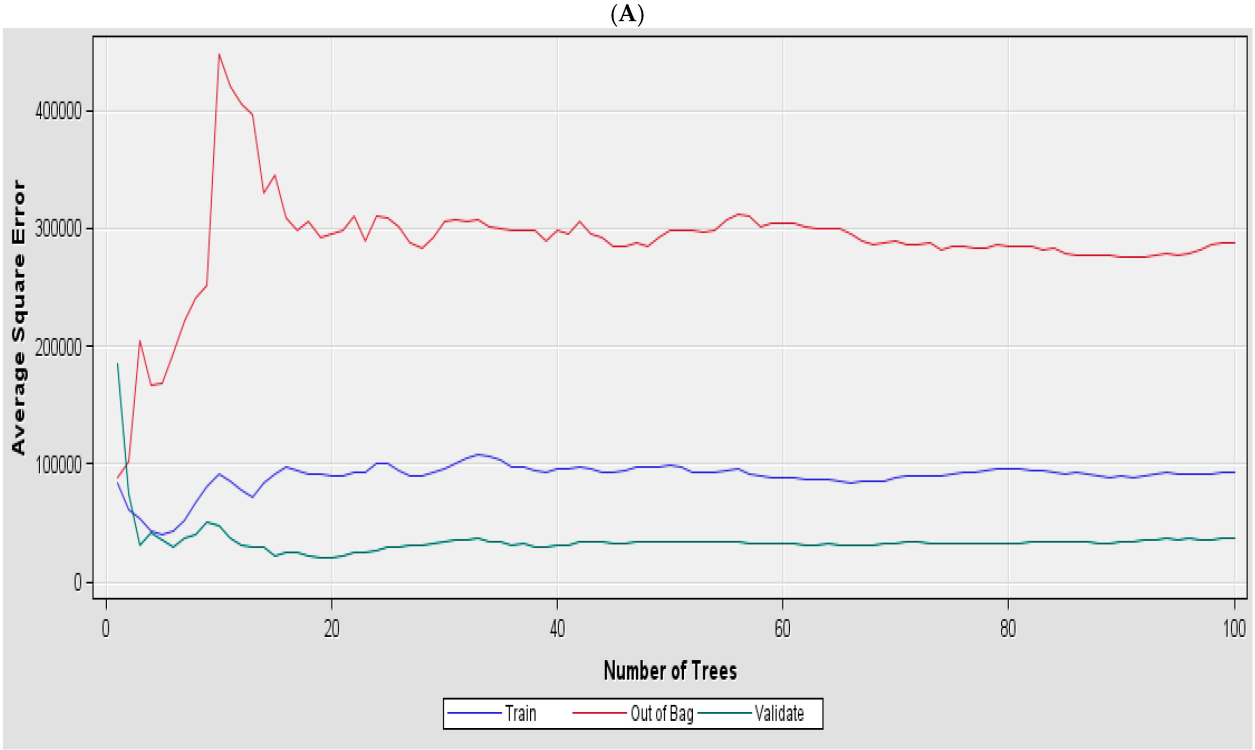
	Decision Tree	**SALQ**
Minimum	−21.1615	−78.3187
Maximum	20.8533	19.543
Mean	0.1394	−1.2729
Standard deviation	3.8397	9.5787

* amounts in USD thousands.

To examine whether the model fits at various levels, the graphs in Figure 4A–C show the mean predicted versus actual quarterly cash tax payments for the decision tree, neural nets, and random forest methods for both training and validation results. Consistent with Table 7, the decision tree method's mean prediction errors were the lowest. Figure 4D shows actual versus predicted quarterly tax payments over time. As can be seen, the two-track well over time.

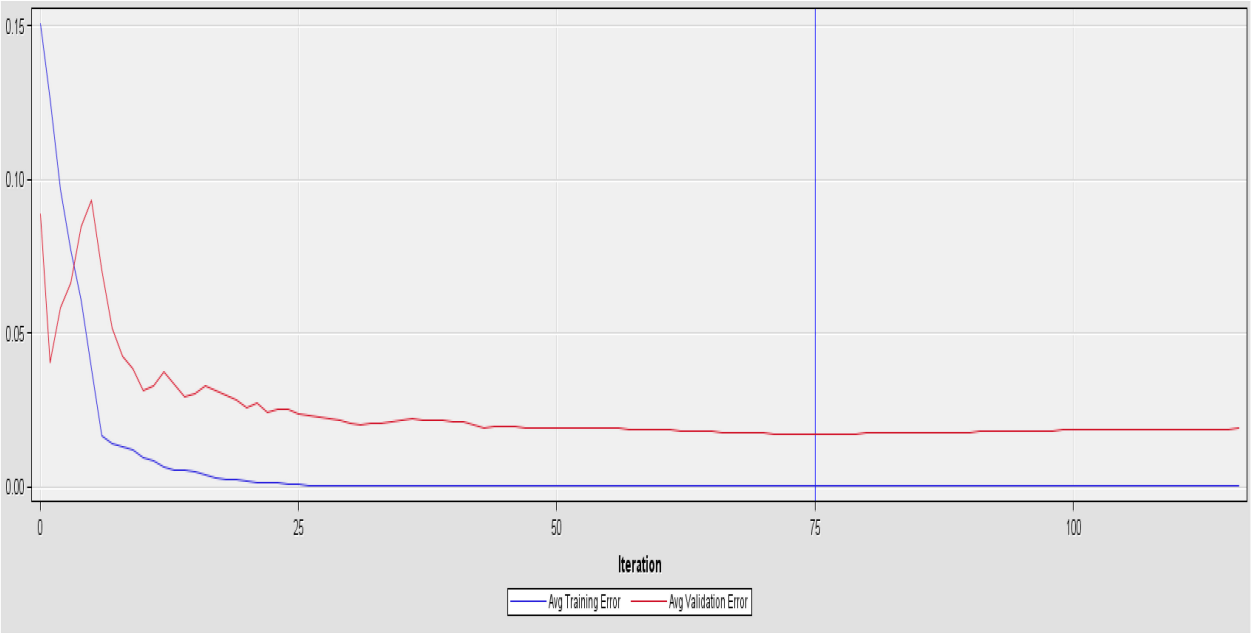


(**A**)

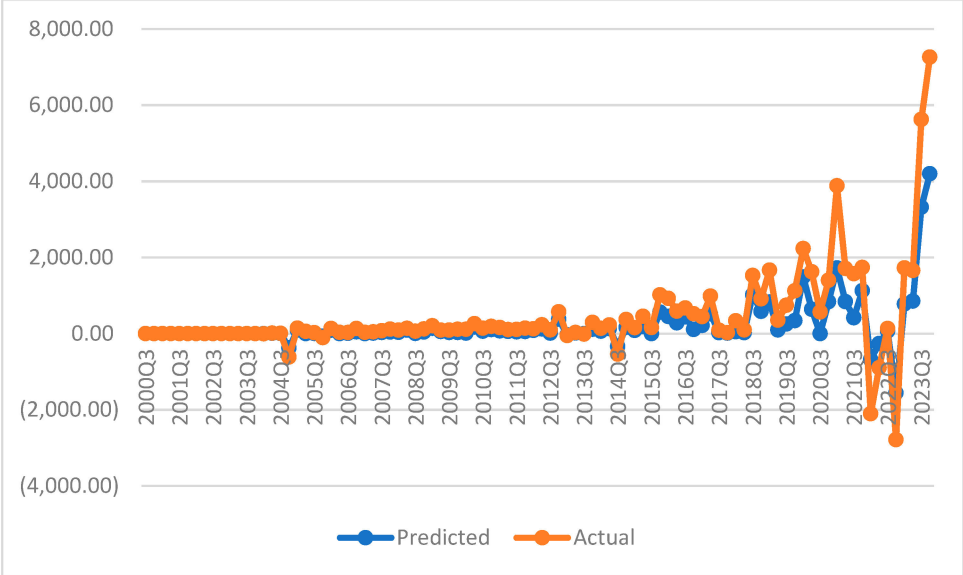


(**B**)

Figure 4. *Cont.*



(**C**)



(**D**)

Figure 4. Amazon: (**A**) decision tree model fits over iterations; (**B**) random forest model fits over iterations; (**C**) neural net model fits over iterations; (**D**) quarterly tax payments-actual versus predicted-over time (USD thousands).

6. Conclusions

This study has demonstrated that deep machine learning methods can provide accurate forecasts of quarterly taxes due by corporations. The results are robust to shocks created by the COVID-19 pandemic economy. The results should be helpful to company managers trying to properly budget such payments and stock analysts trying to forecast companies' after-tax earnings. Increasing availability of statistical packages for use of such methods, without the need for programming skills, should make the tools attractive to managers and analysts (The SAS tool used in this paper's analyses requires no hard coding. Another example is the data manipulation tool Alteryx, which offers R-based "click and drag" deep learning tools). From a policy perspective, more accurate forecasting of tax payments by companies would lead to more predictable tax revenues for the government.

Some potential limitations of the study include the fact that the models presented might not fit all companies (this paper only tested a subsample of publicly traded firms).

Research on machine learning for tax data forecasting could significantly improve tax administration and compliance. By leveraging the power of algorithms and data, researchers can develop models that provide valuable insights and predictions for tax authorities, policymakers, and taxpayers. As technology advances, machine learning in tax forecasting is expected to play a crucial role in enhancing tax administration and revenue collection processes. Challenges in this field include data privacy concerns, the interpretability of machine learning models, and the need for accurate and reliable data. Researchers are addressing these challenges by developing techniques to ensure data security and transparency in machine-learning models for tax forecasting.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

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

References

1. Bratten, B.; Gleason, C.A.; Larocque, S.A.; Mills, L.F. Forecasting Taxes: New Evidence from Analysts. *Account. Rev.* **2017**, *92*, 1–29. [CrossRef]
2. Hirst, D.E.; Koonce, L.; Venkataraman, S. Management Earnings Forecasts: A Review and Framework. *Account. Horizons* **2008**, *22*, 315–338. [CrossRef]
3. Blaufus, K.; Chirvi, M.; Huber, H.-P.; Maiterth, R.; Sureth-Sloane, C. Tax Misperception and its Effects on Decision Making—Literature Review and Behavioral Taxpayer Response Model. *Eur. Account. Rev.* **2022**, *31*, 111–144. [CrossRef]
4. Dixon, M.F.; Halperin, I.; Bilokon, P. *Machine Learning in Finance*; Springer International Publishing: New York, NY, USA, 2020; Volume 1170.
5. Culkin, R.; Das, S.R. Machine learning in finance: The case of deep learning for option pricing. *J. Investig. Manag.* **2017**, *15*, 92–100.
6. Qian, X.Y.; Gao, S. Financial series prediction: Comparison between precision of time series models and machine learning methods. *arXiv* **2017**, arXiv:1706.00948.
7. Milner, C.; Berg, B. Tax analytics—Artificial intelligence and machine learning—Level 5. PWC Advanced Tax Analytics & Innovation. 2017. Available online: https://www.pwc.no/no/publikasjoner/Digitalisering/artificial-intelligence-and-machine-learning-final1.pdf (accessed on 23 September 2024).
8. Neñer, J.; Cardoso, B.-H.F.; Laguna, M.F.; Gonçalves, S.; Iglesias, J.R. Study of taxes, regulations and inequality using machine learning algorithms. *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.* **2022**, *380*, 20210165. [CrossRef]
9. Abedin, M.Z.; Chi, G.; Uddin, M.M.; Satu, S.; Khan, I.; Hajek, P. Tax Default Prediction Using Feature Transformation-Based Machine Learning. *IEEE Access* **2020**, *9*, 19864–19881. [CrossRef]
10. Febriminanto, R.D.; Wasesa, M. Machine Learning Analytics for Predicting Tax Revenue Potential. *Indones. Treas. Rev. J. Perbendaharaan, Keuang. Negara dan Kebijak. Publik* **2022**, *7*, 193–205. [CrossRef]
11. Ippolito, A.; Lozano, A.C.G. Tax Crime Prediction with Machine Learning: A Case Study in the Municipality of São Paulo. In Proceedings of the 22nd International Conference on Enterprise Information Systems, Virtual, 5–7 May 2020; pp. 452–459.
12. Battiston, P.; Gamba, S.; Santoro, A. *Optimizing Tax Administration Policies with Machine Learning*; Management and Statistics Working Paper; Department of Economics, University of Milan Bicocca: Milan, Italy, 2020; Volume 436.
13. Baghdasaryan, V.; Davtyan, H.; Sarikyan, A.; Navasardyan, Z. Improving Tax Audit Efficiency Using Machine Learning: The Role of Taxpayer's Network Data in Fraud Detection. *Appl. Artif. Intell.* **2022**, *36*, 2012002. [CrossRef]
14. Alarie, B.; Niblett, A.; Yoon, A.H. Using machine learning to predict outcomes in tax law. *Can. Bus. LJ* **2016**, *58*, 231. [CrossRef]
15. Adamov, A.Z. Machine learning and advanced analytics in tax fraud detection. In Proceedings of the 2019 IEEE 13th International Conference on Application of Information and Communication Technologies (AICT), Baku, Azerbaijan, 23–25 October 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–5.
16. De Roux, D.; Perez, B.; Moreno, A.; Villamil MD, P.; Figueroa, C. Tax fraud detection for under-reporting declarations using an unsupervised machine learning approach. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; Volume 50, pp. 215–222.
17. Battaglini, M.; Guiso, L.; Lacava, C.; Miller, D.L.; Patacchini, E. Refining public policies with machine learning: The case of tax auditing. *J. Econ.* **2024**, 105847. [CrossRef]
18. Wahab RA, S.R.; Bakar, A. Digital economy tax compliance model in Malaysia using machine learning approach. *Sains Malays.* **2021**, *50*, 2059–2077. [CrossRef]
19. Zumaya, M.; Guerrero, R.; Islas, E.; Pineda, O.; Gershenson, C.; Iñiguez, G.; Pineda, C. Identifying tax evasion in Mexico with tools from network science and machine learning. *Corrupt. Netw. Concepts Appl.* **2021**, 89–113. [CrossRef]

20. Murorunkwere, B.F.; Haughton, D.; Nzabanita, J.; Kipkogei, F.; Kabano, I. Predicting tax fraud using supervised machine learning approach. *Afr. J. Sci. Technol. Innov. Dev.* **2023**, *15*, 731–742. [CrossRef]
21. Lee, C. Training and Interpreting Machine Learning Models: Application in Property Tax Assessment. *Real Estate Manag. Valuat.* **2022**, *30*, 13–22. [CrossRef]
22. Alsadhan, N. A Multi-Module Machine Learning Approach to Detect Tax Fraud. *Comput. Syst. Sci. Eng.* **2023**, *46*, 241–253. [CrossRef]
23. Rahman, R.A.; Masrom, S.; Omar, N.; Zakaria, M. An application of machine learning on corporate tax avoidance detection model. *IAES Int. J. Artif. Intell.* **2020**, *9*, 721–725. [CrossRef]
24. Liu, L.; Xu, H.; Dao, M.; Sun, H. Pilot CEOs and tax avoidance: Evidence from machine learning methods. *Asia-Pacific J. Account. Econ.* **2023**, *30*, 14–29. [CrossRef]
25. Kumar, N.N.; Sridhar, R.; Prasanna, U.U.; Priyanka, G. Opportunities of Machine Learning Algorithms in the Tax Default Prediction-A Survey. In Proceedings of the 2023 7th International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 11–13 April 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1225–1231.
26. Rabasco, M.; Battiston, P. Predicting the deterrence effect of tax audits. A machine learning approach. *Metroeconomica* **2023**, *74*, 531–556. [CrossRef]
27. Lahann, J.; Scheid, M.; Fettke, P. Utilizing Machine Learning Techniques to Reveal VAT Compliance Violations in Accounting Data. In Proceedings of the 2019 IEEE 21st Conference on Business Informatics (CBI), Moscow, Russia, 15–17 July 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–10.
28. Visitpanya, N.; Samanchuen, T. A Machine Learning Approach to Identifying Suspicious Tax Evasion Behavior in Public Financial Data. In Proceedings of the 2023 8th International Conference on Business and Industrial Research (ICBIR), Bangkok, Thailand, 18–19 May 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1152–1158.
29. Didimo, W.; Grilli, L.; Liotta, G.; Menconi, L.; Montecchiani, F.; Pagliuca, D. Combining Network Visualization and Data Mining for Tax Risk Assessment. *IEEE Access* **2020**, *8*, 16073–16086. [CrossRef]
30. Masrom, S.; Rahman, R.A.; Mohamad, M.; Rahman, A.S.A.; Baharun, N. Machine learning of tax avoidance detection based on hybrid metaheuristics algorithms. *IAES Int. J. Artif. Intell.* **2022**, *11*, 1153–1163. [CrossRef]
31. Uddin, N.; Dwianika, A.; Sofia, I.P.; Tchamna, R. Classification of Corporate Tax Compliance in Indonesia Based on k-Nearest Neighbors Algorithm. In Proceedings of the 2023 IEEE World AI IoT Congress (AIIoT), Seattle, WA, USA, 7–10 June 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 297–304.
32. Chen, Y.; Liu, W. The Sentiment Attitude of Weibo Users towards Annual Individual Income Tax Return: Based on Natural Language Processing and Machine Learning Methods. In Proceedings of the 2023 IEEE 6th International Conference on Big Data and Artificial Intelligence (BDAI), Jiaxing, China, 7–9 July 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 67–72.
33. Mi, L.; Dong, B.; Shi, B.; Zheng, Q. A tax evasion detection method based on positive and unlabeled learning with network embedding features. In *Neural Information Processing: 27th International Conference, ICONIP 2020, Bangkok, Thailand, 23–27 November 2020*; Proceedings, Part II 27; Springer International Publishing: Berlin/Heidelberg, Germany, 2020; pp. 140–151.
34. Ebrahimzadeh, A.; Garkaz, M.; Khozein, A.; Maetoofi, A. A novel machine learning-based approach for identification of unrealistic tax returns by eeg signal processing. *Biomed. Eng. Appl. Basis Commun.* **2021**, *34*, 2250007. [CrossRef]
35. Guenther, D.A.; Peterson, K.; Searcy, J.; Williams, B.M. How Useful Are Tax Disclosures in Predicting Effective Tax Rates? A Machine Learning Approach. *Account. Rev.* **2023**, *98*, 297–322. [CrossRef]
36. Molapo, M.A.; Olaomi, J.O.; Ama, N.O. Bayesian Vector Auto-Regression Method as an Alternative Technique for Forecasting South African Tax Revenue. *S. Afr. Bus. Rev.* **2019**, *23*. [CrossRef]
37. Rubin, M.M.; Peters, J.L.; Mantell, N. Revenue forecasting and estimation. In *Handbook on Taxation*; Routledge: London, UK, 2019; pp. 769–800.
38. Hui, X. Comparison and application of logistic regression and support vector machine in tax forecasting. In Proceedings of the 2020 International Signal Processing, Communications and Engineering Management Conference (ISPCEM), IEEE. QC, Canada, 27–29 November 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 48–52.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.