A Machine Learning Approach to Equity Bubble Detection and Market Crash Prediction

Hongkai Yu

Vancouver School of Economics, University of British Columbia

ECON 490

Dr. Jonathan Graves

April 23, 2021

Abstract

In this research, I aim to detect equity bubbles and predict financial crashes in the S&P 500 index. I implement machine learning models with special consideration to the imbalanced data problem. Features of the models are selected based on economic theories and market fundamentals. I find that the Random Forests model with CV-tuned decision threshold performs the best, with a 93.0% balance accuracy rate. The Recurrent Neural Networks with Bidirectional Long-term Short Memory and focal loss function gives unsatisfactory results, possibly due to the modest data size. Features with the highest predictive power are market fundamental indicators, long-term market returns, and the psychology factor. Short-term market returns and macroeconomic indicators are less important for the prediction of market crashes. The model in this research is limited to predicting long-term (more than 3 months) and extreme market downturns.

A Machine Learning Approach to Equity Bubble Detection and Market Crash Prediction

Introduction

Bubbles and market crashes are important themes of financial markets. Asset bubbles describe the situation where asset prices significantly deviate from their fundamental values. Notable historical bubble includes the Dutch tulip mania in 1637, the dot-com bubble in the 1990s, and the US housing bubble in 2000s. Investors who are unaware of the potential risks of bubbles realize huge losses when bubbles burst and the market crashes. Some people believe the market crash in 1982 wiped out the cumulative profits of the history of American banking history (Taleb, 2007).

More recently, despite the general economic downturn caused by the COVID-19 pandemic, stock markets are performing exceedingly well. The S&P 500 index was up by 18.23%, while the tech-heavy NASDAQ gained 43.6% in 2020. With the historical lesson of bubbles, we may wonder if we are experiencing one right now.

Given the catastrophic consequences caused by bubbles, it would be meaningful if we can identify bubbles and predict potential crashes. An accurate bubble detection tool may serve as an early warning system for policymakers or a risk management tool for investors. To build such a bubble detection tool is the primary motivation of this research. While bubbles and market crashes are broad phenomena, in this paper, I focus on bubble detection in the stock markets. Specifically, the scope of the research is limited to the bubble detection in the S&P 500 market index. The main research goal is to come up with a method that can reliably detect equity bubbles and forecast market crashes in the S&P 500 index.

If we can realize the main research goal, we can answer certain further questions. Assuming that the prediction method is transparent enough, we can identify important factors that are the most important for prediction. On the technical side, we can gain insights into the selection empirical methods that work the best for equity bubble detection problems. Also, we can predict if we are in an equity bubble right now using our proposed prediction method.

There is much previous research done in the field of bubble detection and market crash prediction. Some researchers, based on economic theories such as rational choices and fundamental value as discounted dividends, came up with statistical tests like variance bound test for bubble detection (Shiller, 2015). More recently, researchers have applied machine learning methods, which are powerful prediction tools for detecting complex patterns, to the study of predicting market crashes (Chatzis et al., 2018). However, as we will see in the literature review section, previous studies are limited in two ways. First, the theory-based studies are highly reliant on their theoretical assumptions and suffer from problems such as joint hypothesis testing. Second, model-free machine learning studies typically do not include enough input variables. They only include short-term trading variables and have unsatisfactory prediction power.

I attempt to solve these limitations in this paper. To avoid the theory reliance problem, I adopted the definition of bubble purely from observable data. As readers may have noticed, I have used the term "bubble" interchangeably with potential market crashes. That is precisely my definition of bubbles. While it may not be suitable for theoretical debates, e.g., "do bubbles really exist", it captures the practical side of the concept of bubbles for practitioners like policymakers and investors. With this definition, the bubble detection problem is transformed into a supervised machine learning classification problem, where we try to predict the output, i.e., bubble, given the input variables. One of the distinguishing features of this research is that I include more input variables than many previous studies to boost the prediction power of models. I select variables that are suggested by economic theories as input variables for bubble detection, including macroeconomic data, long-term trading trends, and market fundamental indicators. It is worth noting that the validity of my research would not be dependent on the correctness of theories, because they are merely serving as feature-selection tools.

I implemented three machine learning models as candidates for the best classifiers: logistic regression (logit), Random Forests (RF), and Recurrent Neural

Networks with Bidirectional Long Term Short Memory (RNN-BiLSTM). One unique challenge of this classification problem is that the dataset is imbalanced, meaning that there are far more non-bubble instances than bubbles. I have adopted special techniques to mitigate this problem, for example, re-sampling for re-weighting, changing decision thresholds, and using asymmetric loss functions.

The results from the models are satisfactory. My best performing model, RF with CV-tuned decision thresholds (RF-CV), achieves a balanced accuracy of 93.0%. To the best of my knowledge, it is the highest-performing machine learning model for predicting market crashes. I believe this can be used as a useful tool for financial market practitioners.

The rest of the paper is organized as follows. The background section provides a literature review of previous research and my methodology for this research. The data section lists all the input variables, their theoretical justification of relevance, and my definition of the output variable, "bubble." The model section introduces the machine learning models that I selected and the reasons for choosing them. The model section also introduces the methods to mitigate the imbalanced data set problem. The result section presents the model performance and the inference of variable importance. The discussion section diagnoses models, provides a robustness analysis, and discusses the limitation of the research. Finally, the conclusion section summarizes the key findings and the contributions of this research.

Background

TODO

Data

In this section, I will give an overview of the data used in this research. First, I will introduce the input variables, a.k.a., features, and my reasons for selecting them. The features include macroeconomic data, market return trends, and market fundamental indicators. Then, I will explain my definitions of equity bubbles and financial crashes. The range of the data is from January 1960 to December 2020 in the

US. Unless otherwise specified, the frequency of the variable is monthly. The raw data are retrieved from Federal Reserve Economic Data (FRED) and *multpl.com*.

Input variable

GDP growth rate. The real Gross Domestic Product growth rate measures the productivity growth of a nation. It is one of the most important macroeconomic metrics. Given its importance, sophisticated machine learning models might gain insights from the growth pattern for bubble detection. The frequency of the GDP data is quarterly rather than monthly. To make the data frequency consistent, I replicate the quarterly GDP growth rate to all the corresponding months.

Inflation. The growth rate of the Consumer Price Index measures the price inflation in a nation. Similar to the GDP growth rate, it is one of the most important macroeconomic indicators. In the context of financial markets, previous research finds that there is an inverse relationship between inflation and lower share prices (Feldstein, 1978). It might be the case that a high level of inflation is associated with a higher possibility of market crashes.

T-bill yield. The 10-year Treasury constant maturity rate is a key benchmark in the debt market. It captures the long-term expected yield of a risk-free investment. From the perspective of an investor, the debt investment and the equity investment are substitutions. It has been found that bond returns tend to be higher than stock returns when the market uncertainty increases (Connolly et al., 2005). There is reason to believe that inflation could be helpful for detecting equity bubbles.

S&P 500 returns. The Standard & Poor is one of the most widely used stock market indexes. It measures the stock returns of the largest 500 companies listed in the US. As discussed in the following subsection, the output variables — equity bubbles and market crashes are constructed from S&P 500 returns. As a common practice in time-series forecasting, it is natural to use historical returns to forecast future data. To include the price trends in different periods, I include 1-, 3-, 6-, 12-, 60-month returns of S&P 500 as features.

Consumer confidence. As noted by Daniel et al. (2005), the psychological factors of investors have a significant influence on investment decisions. I use the data of consumer opinion surveys as a proxy for the public sentiment of the general business environment. While the capital market sentiment does not always align with consumer confidence, this is the most easily accessible data. Also, unlike other newly invented stock market sentiment indexes, the consumer confidence data is complete between 1960 and 2020.

Shiller P/E ratio. Invented by Robert J. Shiller (Shiller, 2015), the Shiller P/E is also known as the cyclically adjusted price-to-earnings ratio. It measures stock prices as a ratio to the average inflation-adjusted earnings (multpl, n.d.). If the Shiller P/E ratio is too high, it suggests that the stock prices could be overvalued based on the corporate earnings.

Market capitalization-to-GDP ratio. Warren Buffet believes that "the market value of all publicly traded securities as a percentage of the country's business" is a useful metric for investment decisions (Buffett & Loomis, 2001). Given the investment success of Warren Buffet, the effectiveness of this metric seems worthwhile to explore. Since collecting data for all publicly traded securities is cumbersome, I calculate the ratio between the S&P 500 index and the nominal GDP as a proxy. Similar to the real GDP growth rate, the frequency of the nominal GDP is quarterly. I replicate the GDP data to match the monthly frequency.

Table 1 presents a summary of input variables.

An interpretation of the median data in the table is the following: the real GDP growth rate of the current quarter is 3.0% annually; the inflation rate is 3.2% annually; the 10-year Treasury Bond yield is 5.7% annually; the Shiller P/E ratio of the month is calculated to be 20.5; the consumer confidence is 100.5; the market capitalization-to-GDP ratio is 0.1; the S&P 500 returns in the past 1, 3, 6, 12, and 60 months are 1%, 2.4%, 4.5%, 9.7%, and 45.4% respectively.

Variable	Min	q_1	$\tilde{x}$	q_3	Max	$\bar{x}$
real_gdp_growth	-31.4	1.4	3.0	4.7	33.4	3.0
inflation	-19.3	1.5	3.2	5.4	24.0	3.7
tbill_yield	0.6	3.9	5.7	7.7	15.3	6.0
shiller_pe	6.6	15.0	20.5	25.7	44.2	20.6
consumer_confidence	95.7	98.9	100.5	101.1	103.1	100.0
mktcap_gdp_ratio	0.0	0.1	0.1	0.1	0.2	0.1
sp500_return	-20.4	-1.2	1.0	2.8	12.0	0.6
sp500_re3	-31.1	-1.7	2.4	6.2	25.9	2.0
sp500_re6	-37.8	-2.4	4.5	9.9	38.0	4.0
sp500_re12	-42.5	-0.8	9.7	18.3	52.7	8.0
sp500_re60	-32.6	9.4	45.4	70.4	213.9	46.1

Table 1
Summary of features

Output variable

I follow the definition of Chatzis et al. (2018) and define a market crash as an extreme market downturn. An extreme market downturn is an one percentile market return. Given this definition, I define the bubble as a pre-stage of market crashes. The bubble is a binary variable: at a given month can only be in the state of "bubble", denoted as 1, or in the state of "non-bubble", denoted as 0. The operational definition of a bubble is a binary variable that takes the value of 1 if and only if: 1) there will be a market crash in the next six months; 2) the current period does not see a market crash.

The benefit of this bubble definition is that it is entirely based on observable data. It does not involve the theoretical definition of the fundamental value of the stock. As discussed in the background section, definitions of fundamental values are subjected to the assumptions of economic models, and it would limit the power of bubble detection models. Of course, this definition is not suited for a theoretical debate of bubbles; however, it offers practical value for equity market practitioners.

In the above definitions, "one percentile" and "six-month" may seem to be arbitrary choices. Indeed, there is no obvious reason why we should not use "two percentile" and "five months" for the definitions. In the discussion section, I will alter the percentile and period length for a robustness analysis of the model.

Table 2 shows the distribution of output data. We see that non-bubble cases are

much more than bubble cases. This is not unexpected since bubbles are abnormal market phenomenon. However, as we will see in the next section, this imbalanced nature of the output variable would create a problem for modelling.

Figure 1 shows the bubble categories through time. From the graph, we see that our bubble definition correctly identifies most pre-stages of historical market crashes, including the dot-com crash around 2001 and the US housing market crash around 2008. This confirms that our bubble definition is in line with reality.

Variable	**Levels**	**n**	%	∑%
bubble	0	690	94.3	94.3
	1	42	5.7	100.0
	all	732	100.0	

Table 2
Bubble distribution

Finally, as a typical practice for machine learning problems, I randomly split the full data into the training data and the testing data. The training data is 80% of the full data, while the testing data is the remaining 20%.

Model

With the data set up in the previous section, the central question of the research becomes a supervised binary classification problem: we aim to predict the output variable given input variables through machine learning models. In this section, I will give an introduction to the models that I used in this research. Also, I will explain a special modelling challenge in this research — the problem of imbalanced data, and the ways to mitigate the problem.

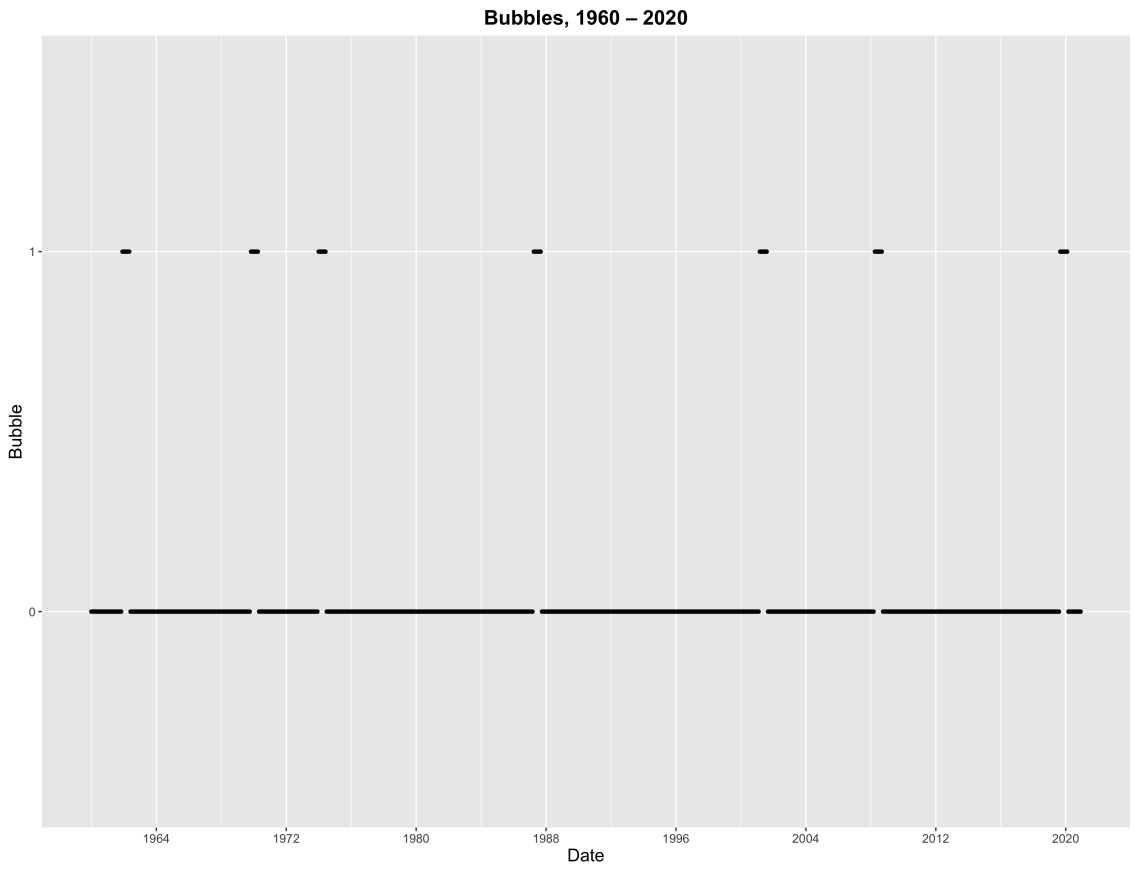
Model selection

In this research, I fit three models to the data: logistic regression (logit), Random Forests (RF), and Recurrent Neural Networks (RNN).

Logistic regression

Logistic regression is one of the most widely used binary classifiers. It is easy to implement, and the interpretation of its results is straightforward. Logistic regression

Figure 1
The definition of bubbles



predicts the probability of the binary output variable through the logistic transformation. Equation 1 shows the formula of the logistic regression, where X is the design matrix of the input variables and Y is the output variable, the bubble. From the formula, we see that the logistic transformation of the probability of bubbles is just the log-odds of bubbles. Therefore, we can also interpret the logistic regression as modelling the log-odds of the bubble using linear models.

$$\text{logit}(P(Y)) = \log \frac{P(Y)}{1 - P(Y)} = X\beta \tag{1}$$

While the logistic regression is easy to implement, it is limited in that it cannot capture the non-linear pattern in the data. The relationship between the input variables and the dependent variables (after the logistic transformation) is restricted to a linear form. However, bubbles and financial markets are complicated, and it is reasonable to assume that the true relationship is more than linearity. This is why I employ more

complex models such as Random Forests and Recurrent Neural Networks to learn the non-linear pattern in the data.

Random Forests

The RF is an ensemble learning method based on decision trees. It fits many decision trees to the data through bootstrap and randomly excluding some features. Then, it aggregates, or "bags" the predictive results of all decision trees to yield the final result. Thanks to the cursive spliting algorithm of the decision tree and the bagging method, the RF is good at detecting non-linear patterns in the data. Besides being a non-linear classifier, the main advantage of the RF is that the bagging mitigates the overfitting problem of the simple decision trees algorithm. Also, the RF is a promising method for finance research. According to a study conducted by W. Lin et al. (2012), soft classification techniques, including ensemble learning methods, “appear to be the direction for future research” in financial crisis prediction. The nature of financial crisis studies are similar to this study. It is reasonable to assume that the techniques which work well for financial crisis studies could also yield good results for this study. As a strong ensemble learning classifier, the RF method is a candidate for the best performing model in this research.

Recurrent Neural Networks

Deep learning models, or artificial neural networks with multiple hidden layers, are becoming popular in finance research. They are often shown to have superior results to traditional methods. The RNN is a family of deep learning methods that are good at predicting sequential natured data, including time-series data. In this research, I choose a specific type of RNNs — RNNs with bidirectional long short-term memory networks (RNN-BiLSTM). The long short-term memory (LSTM) is a neural network unit that helps the model to learn from the long-term pattern in the data. The bidirectional component enables the model to detect patterns in the sequential data in two ways. Readers who are interested in knowing more about how the RNN models work can refer to Olah (2015) and the appendix of Bashchenko and Marchal (2020).

In the context of forecasting financial time series, previous research has shown

that RNNs with BiLSTM generally perform better than traditional time series prediction methods like ARIMA (Siami-Namini et al., 2019). A recent asset bubbles detection study uses this method and gets satisfactory results (Bashchenko & Marchal, 2020). Based on previous studies and the nature of my study, I select this method as a candidate for the best classifier.

The challenge of imbalanced data

Apart from model selection, this research is also faced with a special challenge of imbalanced data. As shown in Table 2, the distribution of the output variable is highly imbalanced. There are way more zeros ("non-bubble") than ones ("bubble"). This imbalanced distribution is expected since bubbles and market crashes are rare , yet it could cause trouble for the modelling.

The default behaviour of a typical machine learning fitting algorithm is to minimize misclassfication cases in all training data. Since the non-bubble case represents the over-majority of the data, without any special treatments, the model fitters are expected to focus more on the non-bubble cases than the bubble cases. This behaviour is undesirable because the research is interested in predicting bubbles. As an extreme case, even a naïve classifier that predicts zero in all cases would have a high accuracy rate. Its prediction is very likely to be correct since it is true that the market does not have bubbles most of the time. However, this kind of naïve classifier is completely useless and offers no practical value. We gain no insights of forecast market crashes using it. To make the research meaningful, we want to make our models less prone to suffer the same problem as naïve predictors.

Besides, from a practical perspective, the loss caused by the failure of predicting a market crash is larger than that caused by a false alarm. The latter may lead to over-conservative investments or policy decisions, but an unexpected market crash can cause huge financial loss and incur big social costs. This practical concern also suggests that the default model-fitting that prioritize predicting non-bubbles offers little practical value, we should focus on correctly predicting the bubbles.

To tackle this challenge, I have tried a couple of ways, including changing

decision thresholds, re-sampling for re-weighting, and adopting an asymmetric loss function. These methods essentially make the models focus more on the bubble cases than the non-bubble cases. No more than one of these methods should be used concurrently; otherwise, there will be an overcorrection. Also, to reflect our preference for the prediction result, I use the balanced accuracy rate rather than the overall accuracy rate as my model evaluation metrics.

Changing the decision thresholds

For all the binary classification methods that I select, they would first predict a probability of the bubble before making a categorical decision. The default decision threshold, as suggested by Baynes' theorem of classification, is 0.5. For example, given a specific set of input data, the model may predict that the probability of bubbles is 0.4. Since the 0.4 is smaller than 0.5, the model would make a categorical prediction of non-bubble.

We can lower the decision threshold to make the model more likely to make a "bubble" decision. Under decreased decision threshold, a lower bubble probability is enough to trigger an alarm of the bubble. If we changed the decision threshold to 0.3, in the above example, the model would give a "bubble" prediction since 0.4 is greater than 0.3. In this way, we make the model treat the bubble probability with higher weights. It can offset the over-representation problem of non-bubble cases.

How to decide the new decision threshold? Setting the threshold to 0.3 seems arbitrary. In this research, I use two methods to determine the new threshold. First, the model can use the prevalence of positive cases i.e., the frequency of bubbles in the training data, as the threshold. This would offset the imbalanced data problem completely. Also, the model can treat the threshold as a hyperparameter and use cross-validation (CV) to tune it. The CV method is particularly suitable for the RF model because the out-of-bag samples of the RF can be easily used for the CV.

Re-sampling for re-weighting

We can also directly change the re-weight in the data through resampling. For each bubble case, we sample the data more than once, while in the non-bubble case we

only sample it once. Essentially, the method replicates the bubble data. We replicate until the ratio between the number of bubble cases and the number of non-bubble cases close to one. Using this method, the distribution of the output variable is perfectly rebalanced.

Asymmetric loss function

A loss function for classification problems is a function that characterizes the misclassification. When fitting a classifier, we minimize the loss function on the training data to make a better prediction. The default loss function for most binary classification algorithms is the binary cross-entropy function. As shown in Equation 2, the loss function is symmetrical for the bubble misclassification and non-bubble misclassification.

$$CE(p) = -\log(p) \tag{2}$$

If we want to make the model focus more on the bubble prediction, we can make the model fitter consider a bubble misclassification as a bigger loss than a non-bubble misclassification. To achieve this, we can adopt an asymmetrical loss function. In this research, I use the focal loss function, which is proposed by T.-Y. Lin et al. (2017). The function takes the form as Equation 3. α and γ are hyperparameters that determine how asymmetric the loss should be and the customized treatments for misclassification losses. I set them as the same as the original paper. This method is most easy to implement in the RNN model since the TensorFlow package, a widely used deep learning library, enables us to use customized loss functions for model fitting.

$$FL(p) = -\alpha(1-p)^{\gamma}\log(p) \tag{3}$$

Model evaluation

I use the balanced accuracy as the model evaluation metric. It is defined as the average value of sensitivity and specificity. Sensitivity measures the proportion of correctly predicted positive (bubble) cases, while specificity measures the proportion of

correctly predicted negative (non-bubble) cases. A formula form of the balanced accuracy is shown in Equation 4.

$$\text{Balanced Accuracy} = \frac{\text{True Negative}}{\text{Negative}} + \frac{\text{True Positive}}{\text{Positive}} \tag{4}$$

This metrics is preferred compared to a simple, overall accuracy rate. It separately measures the model performance on positive (bubble) cases and on negative (non-bubble) cases. Under this metric, a naïve classifier that predicts all cases as non-bubble can only achieve 50% since it has 0 sensitivity.

Results

Prediction

Combining the models and the special consideration to the imbalanced data, I fit five sets of models to the data: logistic regression, using prevalence as the decision threshold (logit-P); logistic regression, using re-weighted data (logit-RW); RF, using CV-tuned decision threshold (RF-CV); RF, using re-weighted data (RF-RW); RNN + BiLSTM, with a focal loss function (RNN-BiLSTM-focal). Their performances are shown in Table 3.

	Model	Sensitivity	Specificity	Balanced Accuracy
1	Logit-P	0.727	0.721	0.724
2	Logit-RW	0.727	0.735	0.731
3	RF-CV	1	0.86	0.93
4	RF-RW	0.455	1	0.727
5	RNN-BiLSTM-focal	0	1	0.5

Table 3
The peformance of models

The RF model with a CV-tuned decision threshold performs the best, having a balanced accuracy of 93%. The sensitivity and the specificity show that the model correctly identifies 100% of the bubble cases and 86% of the non-bubble cases. Most other models perform decently well, with balanced accuracy rates around 70%. The RNN model is a disappointment. It performs no better than a naïve model. I will attempt to explain the difference in the predictive power of the models in the discussion section.

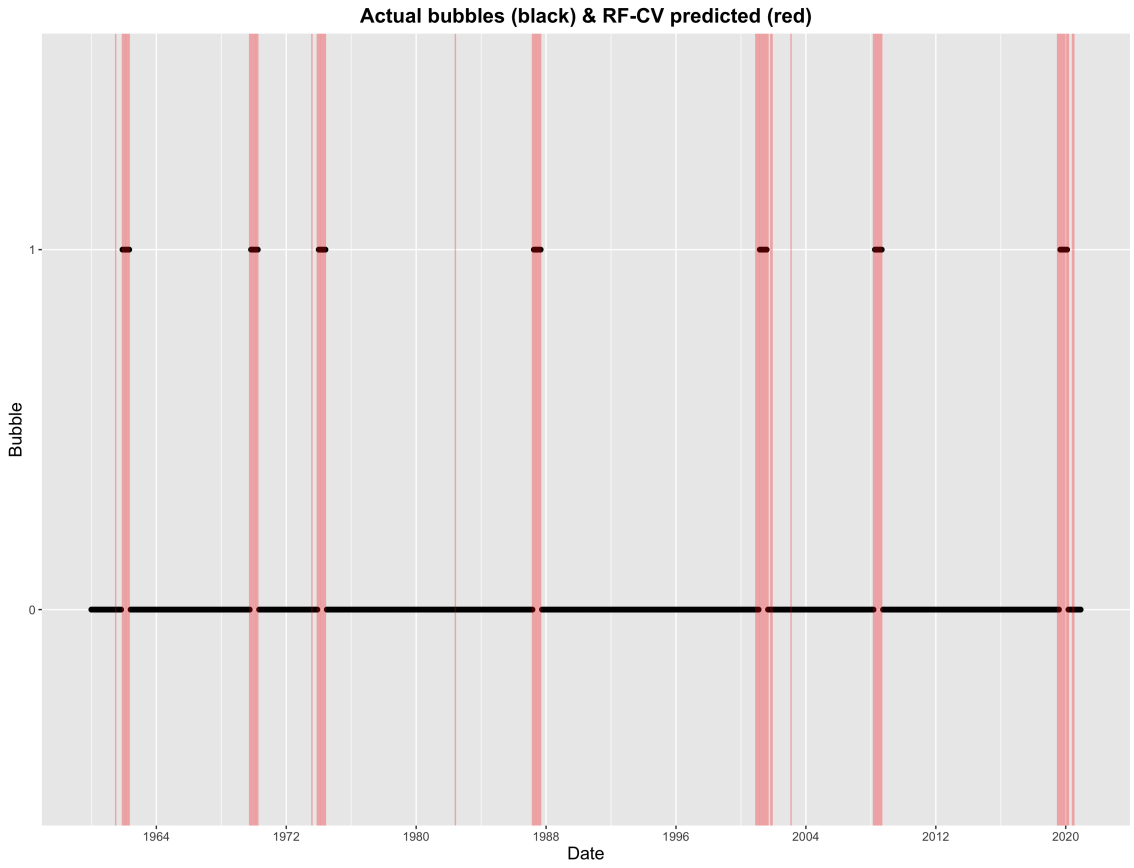
Figure 2 and Figure 3 provide more information about the RF-CV model. Figure 2 shows the process of determining the decision threshold. Cross-validation chooses the decision threshold of 0.08 since it yields the lowest balanced error in the CV data. Figure 3 shows the prediction results (red area) of the RF-CV model when imposed on the complete data set. As we can see, the RF-CV model correctly classifies almost all cases.

Figure 2

The decision threshold tuned by CV

When feeding with in the most recent data (April 2021, the time when this paper is written), the RF-CV model gives the prediction of "bubble." It suggests that we are likely in a bubble, and financial crashes are likely in the next six months. This result is not financial advice, and I do not recommend the reader to make risky financial decisions based on this result. If the reader still wants to take this result as a reference, I highly recommend the reader to read the discussion section for the limitation of this research before making any decisions.

Figure 3
RF-CV prediction of bubbles



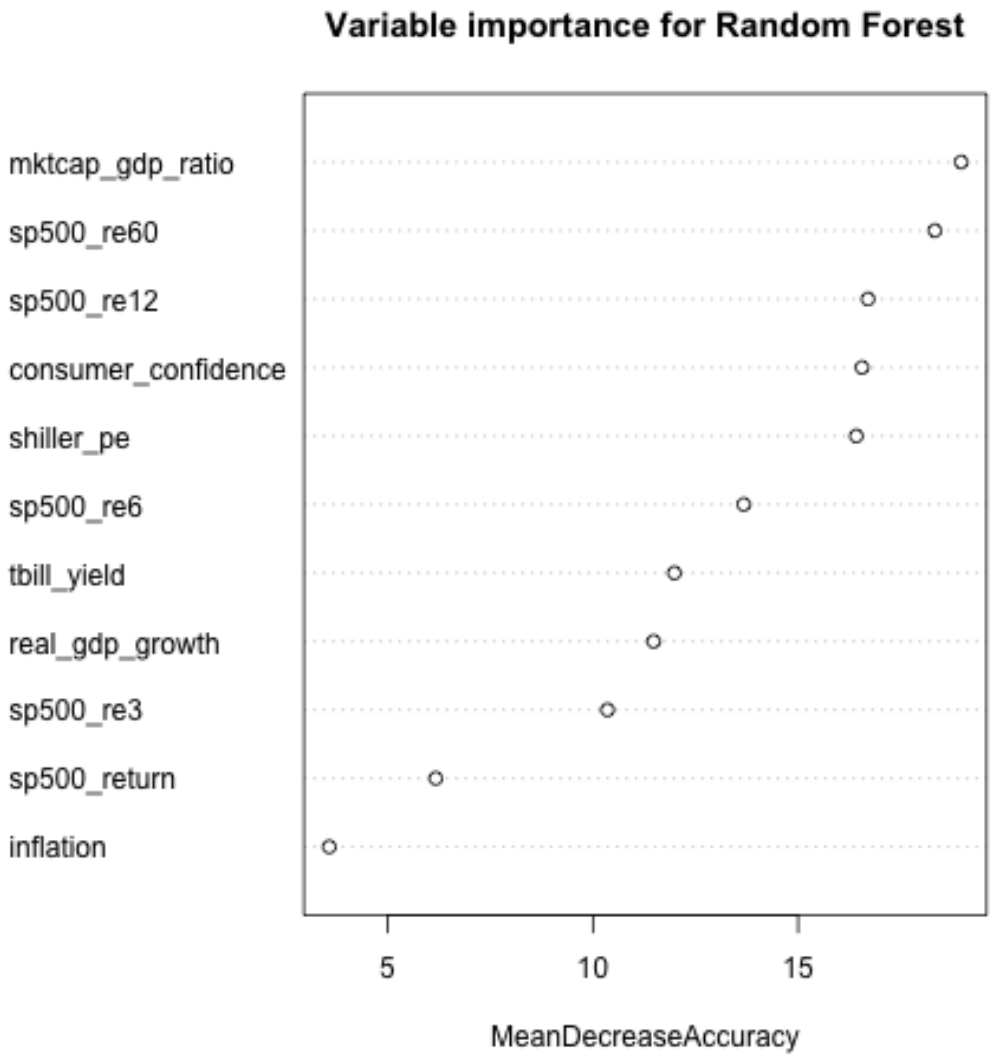
Inference

There are further insights that we can draw from the research besides the prediction results. We can identify important factors for prediction. I use two of the best performing models for inference: the RF-CV model and the logit-RW model.

Figure 4 shows the variable importance in the RF-CV model. We see that consumer confidence, long-term investment returns, e.g., 5-year S&P returns, and fundamental indicators, e.g., market capitalization-to-GDP ratio, are the most important. On the other hand, the short-term S&P 500 return and short-term macroeconomics data are among the least important features.

Table 4 shows the regression table for the logit-RW model. It mostly agrees with the results of the RF-CV variable importance plot. The fundamental indicators and long-term investment returns are significant at the 99.9% confidence level. The 3- and 6- month market returns are not significant. However, contrary to the result of the

Figure 4
The variable importance in the RF-CV model



RF-CV model, the logit-RW model finds inflation and one month market return significant for prediction.

Based on the above inference, it seems that our model favours the investment philosophy held by value investors, in the context of avoiding market crashes. If the stock prices are too high for corporate earnings and national production, it would be wise to heed the risks of market crashes. Also, this inference results explains why the machine learning models from previous studies do not perform well. As mentioned in the background section, most of the previous machine learning studies for predicting financial crashes only include short-term market return data as features. Those

Table 4

	Dependent variable:
	bubble
real_gdp_growth	−0.007
	(0.035)
inflation	0.099***
	(0.028)
tbill_yield	0.237***
	(0.073)
shiller_pe	−0.658***
	(0.058)
consumer_confidence	−1.182***
	(0.147)
mktcap_gdp_ratio	216.592***
	(18.361)
sp500_return	−0.105***
	(0.033)
sp500_re3	0.018
	(0.022)
sp500_re6	−0.021
	(0.020)
sp500_re12	−0.143***
	(0.016)
sp500_re60	0.075***
	(0.006)
Constant	106.314***
	(13.641)
Observations	1,112
Log Likelihood	−410.346
Akaike Inf. Crit.	844.693
Note:	*p<0.1; **p<0.05; ***p<0.01

variables have the lowest predictive power among all the input variables I select.

Discussion

Model diagnosis

Having seen the model performance in Table 3, it is natural to ask why certain models perform better than others. While it is not surprising that the logistic regression models perform less well than more complex models, it is worthwhile to explore why the RF-RW model and the RNN-BiLSTM-focal model do not perform as well as the RF-CV model.

After an investigation of the model performance on the training data, I find that there is an overfitting problem for the RF-RW model. The model performs worse on the testing data than on the training data. The model has a 99.3% balanced accuracy rate on the training data but only 72.7% on the testing data. This implies that the model fits too much on the patterns in the training data.

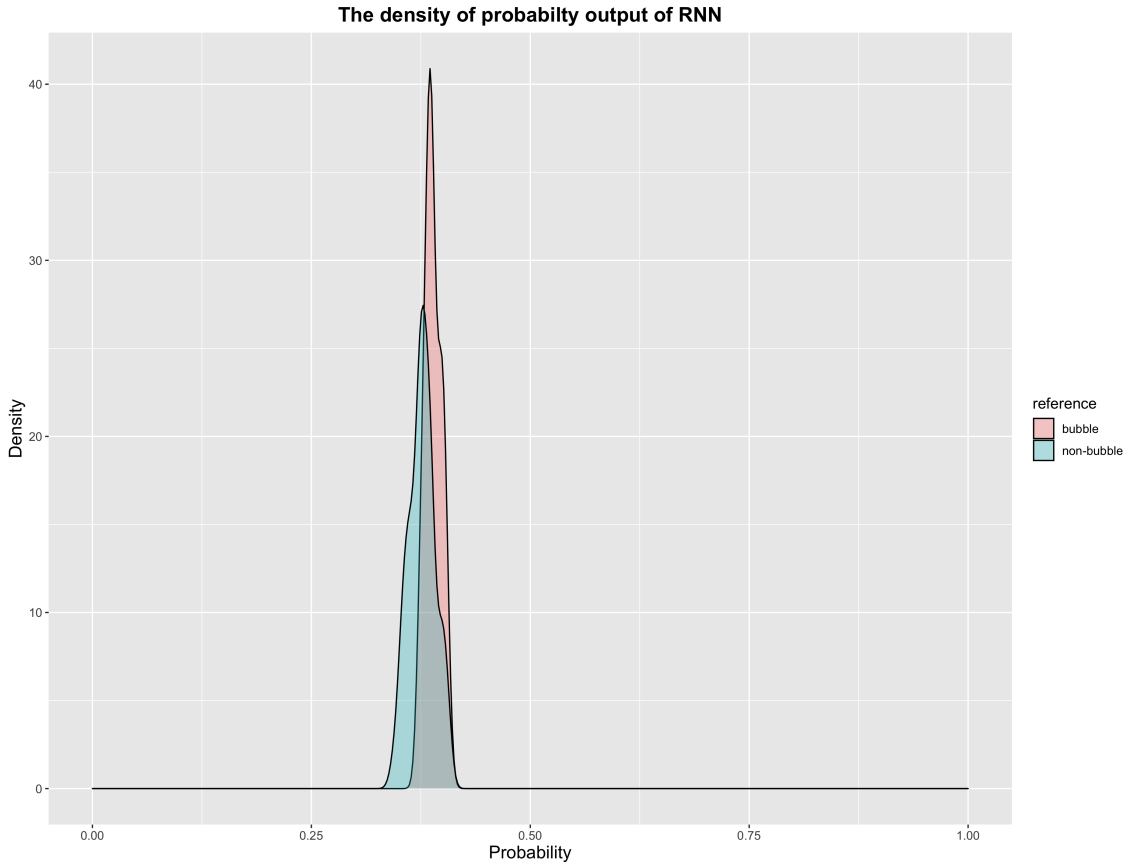
Why would the RF-RW overfit but not the RF-CV? It might be that the re-weighting method makes the model over-trained for the bubble cases in the training data. The re-weighting method replicates the bubble cases, which could cause the model to fit too much on peculiar bubble patterns in the training data. The patterns are not universal for all bubble cases, and they are likely not to be presented in the testing data. This is perhaps why the RF-RW model performs worse on the testing data. On the other hand, the RF-CV only alters the decision threshold. This method makes the model focuses on the bubble cases generally, but not restricted to cases seen in the training data. Without the over-training on the peculiar bubble patterns in the training data, the RF-CV model avoids the overfitting problem.

For the RNN-BiLSTM-focal model, its prediction results are no better than that of a naïve predictor. I have tried various techniques including adjusting hyperparameters, e.g., the number of neurons, changing layers, e.g., using LSTM rather than BiLSTM, and changing the loss function, e.g., changing the hyperparameters of the focal function. None of these techniques works. This is quite a disappointment given the relevant research that endorses the model.

I examined the probability output of the model to analyze the problem. The distribution of the probability output is shown in Figure 5. It shows that the variance of probability outputs of the model is small: almost all outputs are clustered around 0.37. Also, the outputs of bubbles and non-bubbles overlap with each other. The model cannot distinguish the two based on the output. Based on this probability output, I suspect that the model is under-trained. It is likely because the model is too complex for our data size. A study on the impacts of data size on LSTM efficiency finds that three-year daily data is sufficient for training a effective model (Boulmaiz et al., 2020). Three-year daily data is approximately 1000 data points. In this study, we only have 556 data points to train the RNN. It could be that the training data is insufficient to train the RNN model.

Figure 5

The probability output of RNN-BiLSTM-focal



It is worth noting that the above model diagnoses for the RF-RW model and the RNN-BiLSTM-focal model are largely speculative. Due to the complexities of the

models, there does not seem to be an easy way to check my explanations.

Robustness analysis

I did the robustness analysis for the best performing model, the RF-CV model. As discussed in the data section, the quantitative definitions of market crashes and bubbles are rather arbitrary. To check the model performance under different definitions, I alter the "one percentile" and "six months" part of my definition. The results are shown in Table 5.

	0.005	0.01	0.015	0.05
3	0.67	0.88	0.84	0.85
5	0.95	0.95	0.83	0.86
6	1.00	0.93	0.88	0.84
7	0.95	0.95	0.89	0.80
12	0.89	0.85	0.96	0.86

Table 5
Balanced accuracy of RF-CV with different quantiles (column) and periods (row)

We see the model performance is reasonably robust. The model has balance accuracy rates around 95% if the percentile in the market crash definition is lower than 1 and the number of periods in the bubble definition is between 5 and 12. However, the model performs worse for a short-term bubble definition (3-month period) and a less extreme definition of market crashes (5 percentile). This suggests that our model is limited in predicting extreme market crashes in a relatively long timeframe.

The result is not unexpected. The most important factors of the model are market fundamental indicators and long-term trends, so it is plausible that the model based on these factors has less predictive power in the short term. Besides, the method of changing the decision threshold is adopted specifically for the imbalanced data problem. If the data is less imbalanced due to a less extreme market crash definition, this method will not be effective.

Limitations

Besides the limitation that the model is unsuitable for short-term and less extreme market crashes prediction, as discussed in the robustness analysis subsection,

two other limitations are worth mentioning.

The first limitation is regarding the interpretation of the results. A balanced accuracy rate should not be confused with a precision rate. A balanced accuracy rate stands for the balanced percentage of correctly identified cases. On the other hand, a precision rate means among all cases that are predicted as bubbles, the percentage of cases that are actual bubbles. While the RF-CV model has a high balanced accuracy rate, the precision rate is much lower, at 36.7%. This means that if the model gives a "bubble" prediction, there is only a 36.7% possibility that a market crash would take place in the next six months. From a practical perspective, the user of the model should heed the possibility of false alarms when the model predicts bubbles.

Why would the two metrics differ? It can be best illustrated by the confusion matrix of the model prediction. As shown in Table 6, the model correctly identifies most of the cases in both categories, but precision (11/30) is still modest. Although the proportion of misclassified non-bubble cases is low (19/177), the number is non-negligible. Non-bubble cases are much more than bubble cases so that a small portion of misclassified non-bubbles (19) are more than the correctly identified bubbles (11). This drives the precision much lower than the accuracy.

	bubble	non-bubble
bubble	11	19
non-bubble	0	117

Table 6

Confusion matrix of the RF-CV model, reference(columns) and prediction(rows)

The second limitation is that the training data and testing data are not independent. While they are split randomly from the full data, they are both from a similar time period. For example, the data of May 2008 is in the training set, whereas the data of June 2008 could be in the testing set. The two periods are positively correlated due to their proximity in time. At the training stage, the model fitter learns that May 2008 is in a situation of bubble and incorporates its pattern into the model. When faced with the data of June 2008 in the testing set, it is easy for the model to correctly classify it as a bubble case since it has seen a similar instance in the training

data. However, for real-world applications, the model would encounter cases that have much less similarity to the training data. When asked to check if there is a bubble as of April 2021, the model does not have any information about whether March 2021 or May 2021 has a bubble or not. This is a harder problem for the model than the question of June 2008. It is dubious if the model would perform equally well. In short, the dependence between the training data and the testing data is likely to make the testing performance of the models overstated.

Conclusion

I start this research aiming to detect equity bubbles and predict financial crashes. I find that previous empirical studies are limited by their theoretical assumptions or feature selection techniques. To overcome these limitations, I choose the model-free machine learning approach, and the features are selected based on theories but do not rely on theories. Based on the nature of the research problem and relevant studies, I use logistic regression, Random Forests, and Recurrent Neural Networks as my models. To tackle the challenge of the imbalanced output variable, I change the decision thresholds, re-sample the data for weighting, and adopt an asymmetric loss function.

The key findings of the research are as follows. The Random Forest with CV-tuned performs the best, with a balanced accuracy rate of 93.0%. Perhaps due to insufficient data, the RNN model does not give a satisfactory result. Other models perform decently well. It is found that long-term market returns and market fundamental indicators are the most important factors, whereas short-term market returns and macroeconomic data have weak predictive power.

The research has two main contributions. First, I use a new methodology that takes advantage of both the power of machine learning algorithms and the feature-selection insights from economics theories. Second, I build a high-performing predictor (RF-CV) for detecting equity bubbles and predicting market crashes

The research is limited that the model cannot be generalized to short-term and mild market crash prediction. From a practical perspective, the model does not have a high precision rate so that it is prone to send false alarms. Besides, it is suspected that

a positive correlation between the training data and the testing data makes the model overstate its testing performance. Future research is needed to analyze deeper and overcome the correlation problem.

References

Bashchenko, O., & Marchal, A. (2020). Deep learning for asset bubbles detection. *Capital Markets: Market Efficiency eJournal*.

Boulmaiz, T., Guermoui, M., & Boutaghane, H. (2020). Impact of training data size on the lstm performances for rainfall–runoff modeling. *Modeling Earth Systems and Environment*, *6*, 2153–2164.

Buffett, W., & Loomis, C. (2001). Warren buffett on the stock market. *Fortune Investor's Guide*, 80–94.

Chatzis, S. P., Siakoulis, V., Petropoulos, A., Stavroulakis, E., & Vlachogiannakis, N. (2018). Forecasting stock market crisis events using deep and statistical machine learning techniques. *Expert Systems with Applications*, *112*, 353–371. https://doi.org/https://doi.org/10.1016/j.eswa.2018.06.032

Connolly, R., Stivers, C., & Sun, L. (2005). Stock market uncertainty and the stock-bond return relation. *Journal of Financial and Quantitative Analysis*, 161–194.

Daniel, K., Hirshleifer, D., & Subrahmanyam, A. (2005). Investor psychology and security market under-and overreaction. *Advances in Behavioral Finance, Volume II*, 460–501.

Feldstein, M. (1978). *Inflation and the stock market* (Working Paper No. 276). National Bureau of Economic Research. https://doi.org/10.3386/w0276

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection, In *Proceedings of the ieee international conference on computer vision*.

Lin, W., Hu, Y., & Tsai, C. (2012). Machine learning in financial crisis prediction: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *42*(4), 421–436. https://doi.org/10.1109/TSMCC.2011.2170420

multpl. (n.d.). Shiller pe ratio [(Accessed on 04/17/2021)].

Olah, C. (2015). Understanding lstm networks. http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Shiller, R. (2015). *Irrational exuberance* (3rd ed.). Princeton University Press. https://EconPapers.repec.org/RePEc:pup:pbooks:10421

Siami-Namini, S., Tavakoli, N., & Namin, A. S. (2019). A comparative analysis of forecasting financial time series using arima, lstm, and bilstm. *ArXiv*, *abs/1911.09512*.

Taleb, N. N. (2007). *The black swan: The impact of the highly improbable* (Vol. 2). Random house.