Market Regime Determination and Prediction

***Abstract*- This paper studies the determination and prediction of market regimes based on the Russell 3000 index. We use various machine learning techniques K-Means, Self-Organizing Map, Gaussian Mixture Model, Markov Regime Switching Model, and Hidden Markov Model, to delineate market regimes from 1998 to 2017. Then we choose the best clustering model and use its labels to predict the near future state of the market through XGBoost and LSTM. To verify the validity of our approach, we come up with several trading strategies based on our interpretation of the market against a buy-and-hold strategy for the years 2018-2021.**

***Keywords*- Russell 3000 index, market state modeling, machine learning, feature engineering, trading strategy**

I. INTRODUCTION

Financial markets tend to change their behavior over time, which creates booms and busts. Investors often look to discern the current market regime to make macroeconomically aware investment decisions[3]. However, unlike prices, market regimes are not directly observable and hard to determine. In this paper, we apply unsupervised machine learning models to cluster the market regime of the historical period and then use supervised models to predict the state of the near future market. The ideal market regimes we strive to distinguish are long-term, persistent states that can inform trading decisions.

II. DATA SELECTION

Price series of the Russell 3000 index between 1998 and 2017 are chosen to help us test the period between 2018 and 2021. We keep the training period long enough, so it captures major market turning points such as the dot-com bubble and the financial crisis of 2007-2008. Weekly price series instead of daily series are chosen so that we can focus on the predominant longer trend while ignoring the noise and volatility of day-to-day fluctuations.

Stock market is self-organized by the actions of all traders, many of whom use technical analysis to make conclusions about the underlying stochastic price series. Although technical indicators have their names and interpretations, they all fit into one of the three categories: trend, momentum, and volatility[1]. In order not to overfit the training period, we select one or two indicators from each category to help us distinguish market regimes. Below are the six technical indicators we select to help us detect the market states: bear, bull, or static.

A. Trend

Commodity Channel Index: The CCI measures the difference between a security's price change and its average price change. High positive readings indicate the prices are above their average, which is a show of strength. However, the indicator becomes overbought or oversold when it reaches a relative extreme. Therefore, by adding CCI, we hope to catch cyclical patterns of the market and trend reversals.

Mass Index: The calculation of CCI is solely based on the close price series. However, weekly highs and lows are important indicators of a stock's current value and future movement. By examining the range between high and low stock prices over a period of time, a mass index suggests that a reversal of the current trend will likely occur when the range widens beyond a certain point and then contracts.

B. Momentum

Relative Strength Index: RSI measures the magnitude of recent price changes to evaluate overbought and oversold conditions. Traditional interpretations of the RSI are that values above 70 indicate a security is overbought and may be primed for a trend reversal or corrective pullback. An RSI reading below 30 indicates an oversold or undervalued condition.

C. Volatility

Donchian Channel: The Donchian channel is formed by taking the highest high and the lowest low of the last n periods. It measures the volatility of a market price: if a price is stable, the Donchian channel will be relatively narrow. If the price fluctuates a lot, the Donchian channel will be wider.

D. Return

Past One-Month/Three-Month Returns: We add a fourth category to the list since returns are the most directly observable factor of price movements. Past one-month returns and three-month returns are picked to smooth the fluctuations of returns while enabling us to detect the major turning points of the market.

III. DETERMINATION MODELS

In the following section, we test five clustering models for market regime detection, using candlestick visualization and a number of statistics as performance measurements.

A. K-means is a centroid-based clustering model that iteratively assigns data points to a number of clusters. Its main goal is to minimize the sum of distances between the points and their respective cluster centroid. We decide the optimal number of clusters $k = 6$ using the Elbow method and the Silhouette Score. As shown in Figure 1, we generate six temporal sequences by compounding weekly returns of each cluster and group them into three regimes by dynamic time warping (DTW), an algorithm used to measure the similarity between the temporal sequences. A shorter distance stands for higher similarity.

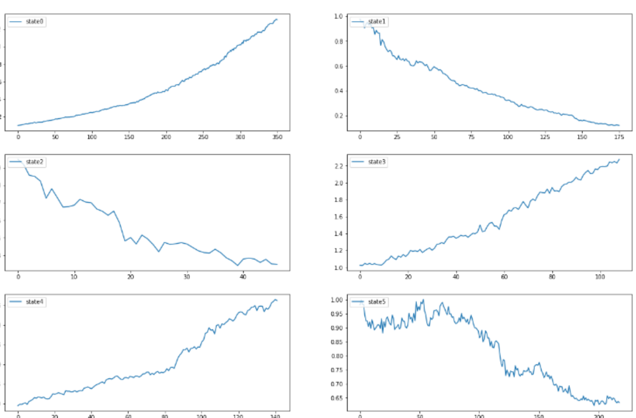

Fig. 1: Six Temporal Sequences of K-means

However, the clustering result shows that generated states are over-dispersed, while our ideal marker regime should be prevalent and change less frequently. In addition, K-means is very sensitive to initial seeds and produces different clustering results every time, which makes the model extremely unreliable.

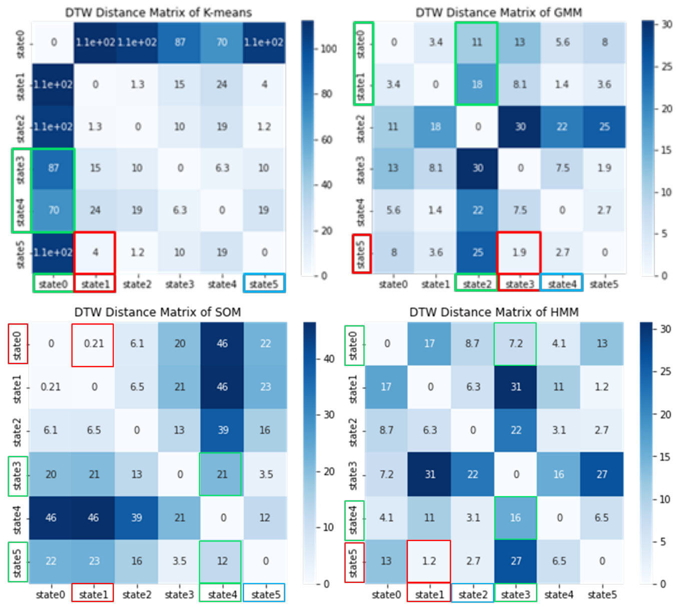

Fig. 2: DTW Distance Matrix

B. Self-Organizing Maps (SOM) is an unsupervised neural network commonly used for high-dimensional data clustering with a competitive learning algorithm, rather than the error-correction learning used by other artificial neural networks. Neuron weights are adjusted based on their proximity to "winning" neurons (i.e., neurons that most closely resemble a sample input). After several iterations, similar neurons are grouped and vice versa. Compared with K-means clustering, SOM has better performance in detecting the bear state but the clustering result is still scattered, especially during the static states. Both SOM and K-means are discriminative models, which directly discriminate the observed data regardless of their underlying distribution. Hence, building a generative model like Gaussian Mixture Models may make improvements.

C. Gaussian Mixture Models (GMM) is a distribution-based clustering model that assumes all data points are generated from a mixture of a finite number of Gaussian distributions. Bayesian Information Criterion (BIC) is a criterion for model selection among a finite set of models. Models with lower BIC are generally preferred. Accordingly, we select $k = 6$ as the optimal number of clusters and perform grouping by DTW. Our clustering result shows the detection of bear and bull regimes is more reasonable and less dispersed, but it incorrectly marks the obvious bull market as static after 2013.

Moreover, the above three models all have the same problem of ignoring temporal regime dependency. To address the issue, we introduce the Markov Model that knows its current state and learns from given inputs to switch among regimes stochastically.

D. Three-State Variance Switching Model

Kim, Nelson, and Startz's three-state variance switching model[4] provides a potential solution to our goal of determining market regimes based on return variances with no mean effects. Historically, bull markets are typically characterized by slow and steady gains that occur without much variation. Bear markets, on the other hand, can be extremely volatile. Static markets may have the lowest variance among the three regimes, or they may be more volatile than bull markets but just fluctuate up and down. However, we found two problems during testing.

First, there is a switch between the low-variance and the medium-variance regimes on February 2nd, 2018. Since our testing period is between 2018 and 2021, it would be difficult to justify our regime decision at that moment.

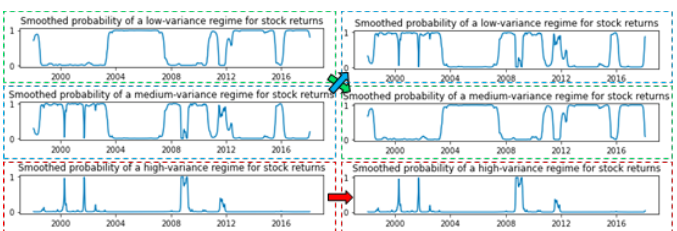

Fig. 3: Variance Regime Switch Problem

Furthermore, based on the clustering of our training period, the three-state variance switching model seems to work fine after 2003. But it fails to determine the correct market states before and after the dot-com bubble. Given the ever-changing nature of the market, a more robust model that supports inputs other than return variances is still needed.

E. HIDDEN MARKOV MODEL (HMM)

HMM seeks to recover a sequence of "hidden" states from the observed data, named "emissions".

An HMM is specified by the following variables[2]:

- Hidden states $\mathbb{S} = \{s_1, s_2, \dots, s_N\}$. $\mathbb{S}$ is the set of all possible states. N is the number of hidden states. Let q_t, the value of hidden state at time t.
- Observation sequence $X = \{x_1, x_2, \dots, x_T\}$. T is the number of observations.
- Transition probability $A = \{a_{ij}\}$ where:
  $$a_{ij} = P(q_{t+1} = s_j | q_t = s_i)$$
  $$1 < i, j < N$$
- Emission probability $B = \{b_j(m)\}$ where:
  $$b_j(m) = P(X_t = x_m | q_t = s_i)$$
  $$1 < j < N, 1 < m < T$$
- Initial state probability $\pi = \pi_i$ where:
  $$\pi_i = P(q_1 = s_i)$$
  $$1 < i < N$$
- Hidden Markov Model will be set as: (A, B, π)

We use BIC to help us determine a range of potential best numbers of hidden states. Figure 4 shows that the BIC curve approximately reaches its lowest level after five components.

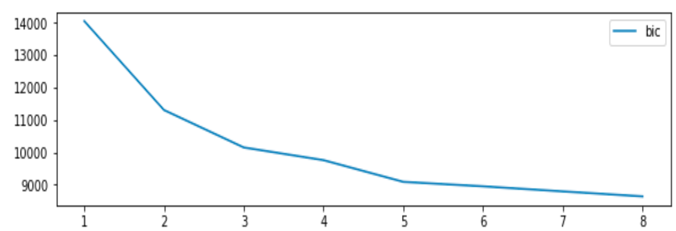

Fig. 4: K Selection by BIC

We divide the dataset from 1998 to 2021 into three parts. 1998-2013 is used as training data. 2013-2018 is the validation set. 2018-2021 is left for backtesting. After experimenting and combining states with DTW, we choose six as the optimal number of states since it generates the best candlestick chart visualization without overfitting.

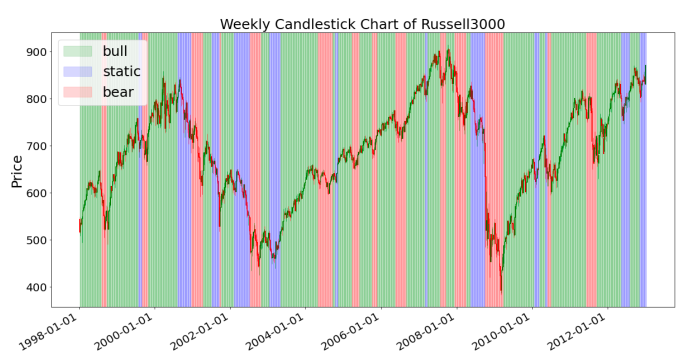

Fig. 5: Candlestick Chart with Detected Regimes of Training

HMM clusters the training period into Figure 5. Unlike the clustering of previous models, each state has a relatively stable duration before the regime changes. Additionally, we present the transition matrix in Table 1. The values on the diagonal, which stands for the probability that the current state will stay the same to the next period, are larger than anywhere else. This result further demonstrates why HMM can produce these continuous states.

Parameter estimation of 6-state HMM						
Initial state probabilities	P1	P2	P3	P4	P5	P6
	0.00	0.00	0.00	0.00	1.00	0.00
Transition probabilities	to S1	to S2	to S3	to S4	to S5	to S6
From S1	0.91	0.03	0.01	0.01	0.00	0.04
From S2	0.03	0.94	0.00	0.01	0.00	0.03
From S3	0.00	0.00	0.95	0.00	0.05	0.00
From S4	0.04	0.04	0.00	0.92	0.01	0.00
From S5	0.01	0.00	0.00	0.06	0.91	0.01
From S6	0.00	0.03	0.01	0.00	0.06	0.90
State mean and std	S1	S2	S3	S4	S5	S6
Mean	-0.02	0.02	-0.06	0.02	0.03	-0.02
Std	0.04	0.02	0.10	0.03	0.05	0.05

Table 1: Transition Matrix

HMM switches its hidden states based on observations. Therefore, our six indicators should have dramatically different distributions under each state. Before applying the model to the testing data, we use the radar charts to analyze the mean and variance of the features to confirm our thought.

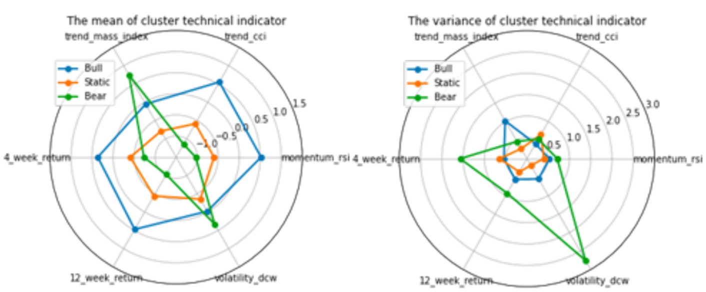

Fig. 6: Radars Chart of Indicators Mean and Variance

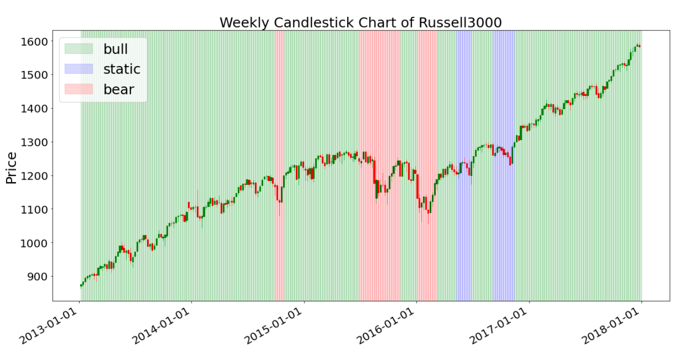

Fig. 7: Candlestick Chart with Detected Regimes of Validation

Figure 7 shows the clustering of the validation period. To verify its correctness, we need to check for the mean returns of each state. In addition, according to the three-state variance switching model, each state should have a distinct distribution of volatilities. By applying the Student's t-test on Donchian channel bandwidth, we fail to prove that the volatility means of each state are significantly different from one another.

Bonferroni corrected α = 0.016 Green cells means the pair is different	t-test		
p-value	Bull	Static	Bear
Bull		0.04022	0.614863
Static	0.04022		0.313582
Bear	0.614863	0.313582	
Mean and std of volatility	Bull	Static	Bear
Mean	-0.75	-0.58	-0.71
Std	0.33	0.35	0.49
Mean and std of return	Bull	Static	Bear
Mean	0.003	-0.001	-0.001
Std	0.013	0.016	0.025

Table 2: T-test Table of Validation

IV. PREDICTION MODELS

Our Hidden Markov Model can be applied to determine the state of the market, but we would not know the actual state until the end of a week since the model uses weekly data. However, regime switches can happen on any day of a week. For example, a stock market crash usually comes with a sudden dramatic decline in stock prices. By the time the week is clustered as bearish, we might already endure a significant loss of paper wealth. Therefore, it is crucial to forecast the output state of HMM.

A. Extreme Gradient Boosting (XGBoost)

XGBoost provides an implementation of gradient boosted decision trees (GBDT). GBDT creates a sequence of decision trees by iteratively training an ensemble of shallow decision trees, with each iteration using the error residuals of the previous model to fit the next model. Unlike most GBDT algorithms, which train the decision trees in sequence, XGBoost has the advantage of doing it in parallel to save computation time. Moreover, XGBoost has built-in parameters for regularization and cross-validation to ensure both bias and variance are kept at a minimum.

By utilizing the built-in feature importance function in XGBoost, we select the final set of features among the determined state from last week and multiple technical indicators. We train 1,000 decision trees in our model with a maximum tree depth set to six. To illustrate, we present the first three layers of our first decision tree.

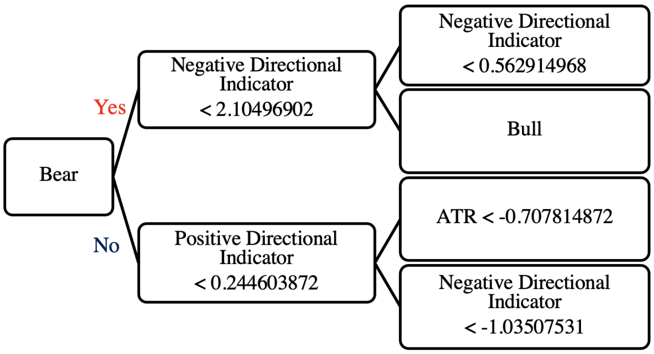

Fig. 8: First Three Layers of First Decision Tree

According to Figure 8, the tree starts with whether the previous state is bear or not. Then, the positive and negative directional indicators are used to decide if the state will change. It makes sense because directional indicators could evaluate the state direction and state strength. The following questions involve Average True Range (ATR), a volatility measurement. Since a bear market usually accompanies high volatility index performance, ATR might imply an upcoming bear state when it goes high.

B. Long Short-Term Memory (LSTM)

LSTM is a particular type of Recurrent Neural Network that is well-suited to learn order dependency in sequence prediction. In general, LSTM[7] consists of three parts: forget gate, memory gate, and output gate. The forget gate decides whether the information from the previous timestamp is relevant and needs to be remembered. The memory gate quantifies the importance of the new information carried by the input. The output gate passes the updated information from the current timestamp to the next.

As shown by Figure 9, our second classification model is a deep neural network with five layers. The first layer takes the past 20 days' daily return, the Donchian channel that helps us capture daily highs and lows, and the binary encoded last week's state as inputs. Due to the neural network's capability to discover and model non-linear and complex relationships, we decide not to provide additional indicators as inputs and rely on the model to learn from the price movements. After the input layer, an LSTM layer is added to capture the time dependency. To prevent overfitting, a dropout layer is inserted between the LSTM layer and the first dense layer, which allows the temporal features identified by LSTM to be recombined in any way optimal to predict the output of HMM. Finally, a second dense layer is added to make the prediction.

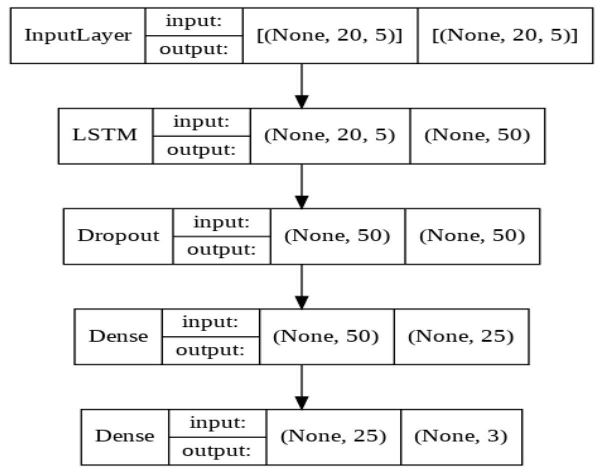

Fig. 9: Summary Graph of LSTM

C. Evaluation of XGBoost and LSTM

Assuming that HMM always generates the true state of the market at the end of a week, we utilize confusion matrices to evaluate the performance of our prediction models. Based on Figure 10, XGBoost achieves 90.5% predicting accuracy, while LSTM has 96.0% accuracy. Since the U.S. equity market possesses the property of long-term bull states and negatively skewed returns, a critical factor to compare models is whether the model could correctly predict the bear state.

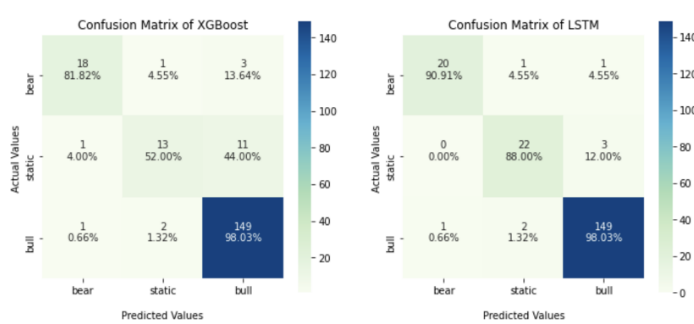

Fig. 10: Confusion matrixes of XGBoost and LSTM

Table 3 shows that LSTM has a better bear-state prediction than XGBoost, indicating that using LSTM could lead to prompt position rebalance.

	XGBoost	LSTM
Precision	0.900	0.952
Recall	0.818	0.909
F1-score	0.857	0.930

Table 3: Performance Comparison for Bear-State Prediction

V. BACKTEST

We design a simple switching strategy based on the market regimes defined by us. We will long Russell 3000 if the market state is bull, short if the market is bear, and buy 3-month T-Bill bonds if the market is static. The latest 3-month T-Bill bond rate is around 0.35%. Noticing a declining trend of the 3-month T-Bill bond rates, we choose 0.30% as our bond rate in the backtest to be conservative.

A. Determination

First, we prove the robustness of our regime determination process. We make weekly investment decisions based on the market regime of that week and compare its performance against the buy-and-hold strategy.

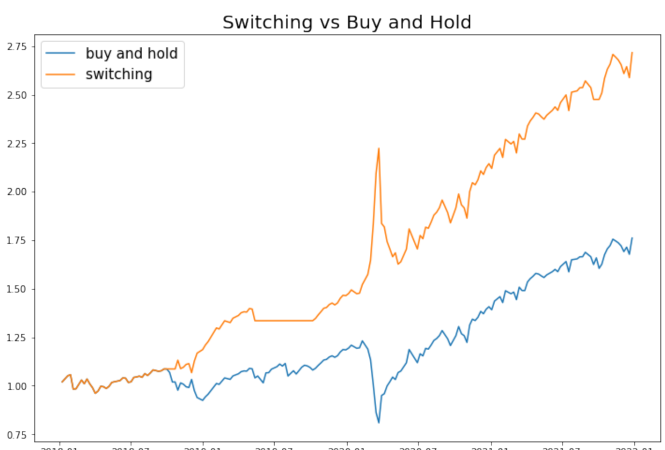

Fig.11: Cumulative Return of Switching Strategy on Determined Regimes vs. Buy-and-Hold Strategy

Strategy	Cumulative Return	Sharpe ratio	Omega ratio	Max drawdown	Calmar ratio
Buy-and-Hold	1.762	0.77	1.17	-35.09%	0.44
Switching	2.763	1.35	1.35	-26.74%	1.11

Table 4: Performance of Switching Strategy on Determined Regimes vs. Buy-and-Hold Strategy

According to Figure 11, the cumulative return of the switching strategy is much higher than that of the buy-and-hold strategy. Table 4, with a list of key performance indexes, also suggests that the switching strategy is more stable and able to achieve a higher risk-adjusted return when compared with buy-and-hold. Hence, we conclude that applying the switching strategy on top of the market states defined by HMM outperforms the buy-and-hold strategy, verifying the validity of our determination.

B. Prediction

Second, we test the performance of our LSTM regime prediction. According to Figure 12, the switching strategy performs well before the stock market crash in February 2020 but doesn't react quickly when the market begins to rebound. As a result, its overall risk-adjusted return and stability are worse than those of the buy-and-hold strategy. New features are needed so that our HMM model can make timelier detection of the market trend reversal.

Fig.12: Cumulative Return of Switching Strategy on Predicted Regimes vs. Buy-and-Hold Strategy

Strategy	Cumulative Return	Sharpe ratio	Omega ratio	Max drawdown	Calmar ratio
Buy-and-Hold	1.762	0.77	1.17	-35.09%	0.44
Switching	1.392	0.50	1.12	-36.77%	0.24

Table 5: Performance of Switching Strategy on Predicted Regimes vs. Buy-and-Hold Strategy

VI. FEATURE ENGINEERING

A. Feature Extraction

We consider constructing an additional feature pool from the following dimensions: stock indexes, derivatives market, bond market, currency market, commodity, macroeconomic conditions, and non-price-related technical indicators. 1) Stock indexes play an essential part in the overall analysis of the U.S equity market. 2) The derivatives market captures investors' bets on future market movements. 3) Bond market rates, such as the treasury yield curves, have long been used as leading indicators of economic growth. 4) Both the currency and commodity markets can imply the market's risk aversion. 5) Macroeconomic conditions can be useful to diagnose the economy's overall health and help us catch longer-term trends. 6) Non-price-related technical indicators of Russell 3000 will provide more information about the index's evolution. All feature data are extracted from the CBOE website and Bloomberg terminal.

B. Data Cleaning

As a prerequisite, datasets with missing values require data cleaning. Since the number of missing values in our datasets is small and the skewness of each feature is low, we apply mean imputation to the missing data. In addition, we use forward filling to deal with the missing rows of lagging indicators.

C. Feature Selection

1) *Feature Correlation Analysis:*

According to the Kolmogorov Smirnov Test, none of our features follow a normal distribution. Hence, Spearman's rank correlation is adopted to describe the inherent relationships between features and Russell 3000's return instead of the Pearson correlation coefficient.

After dropping features having a low correlation with the Russell 3000's one-month return or high correlation with other features, we reduce our feature pool to the ones in Figure 13.

2) *Laplacian Score:*

We also apply the Laplacian score[5], an unsupervised feature selection filter, to evaluate the feature importance and help us capture potentially important features dropped by feature correlation analysis.

The value of the Laplacian score stands for the importance of the corresponding feature. Based on the results of feature correlation analysis and Laplacian scores from Figure 14, we suggest the following five features for HMM: Percentage of members with price above 50-day moving average, VIX index, equity put/call ratio, gold/oil, and 1-year yield. After fitting all combinations of these five features into the model, we conclude that adding Percentage of members with price above the 50-day moving average and VIX index enhances model performance the most.

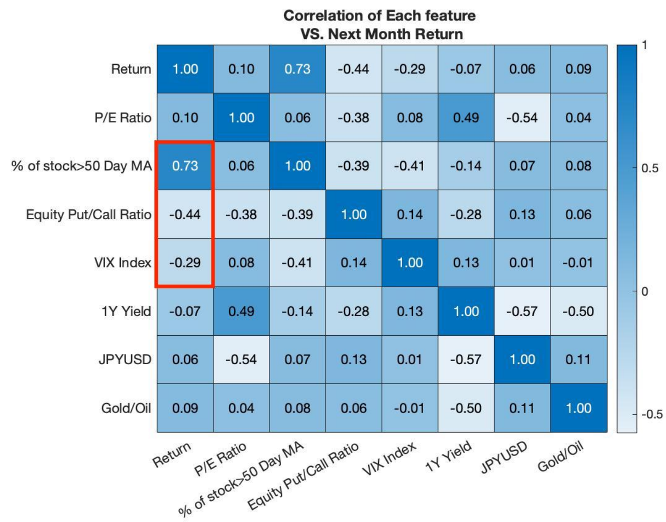

Fig. 13: Correlation of Each Feature vs. Next One Month Return of Russell 3000

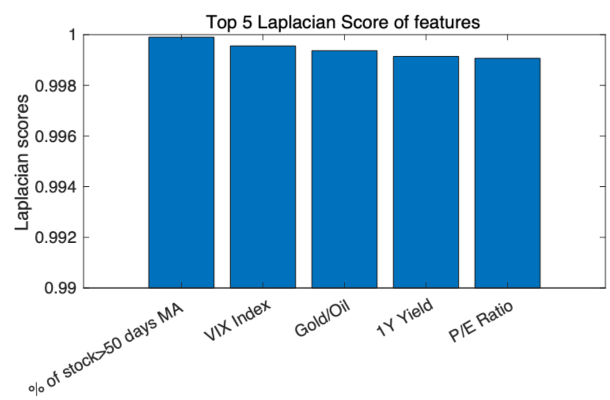

Fig. 14: Top 5 Laplacian Score of Features

VII. MODELS IMPROVEMENT AND REPEAT BACKTESTING

A. Updated HMM and LSTM with New Features

We examine the effectiveness of the new features by repeating the previous analysis steps. First, our updated HMM model is used to cluster the market regimes of the validation period. The same mean analysis of returns and Student's t-test of volatility are applied to examine the three clusters. Based on Table 6, the mean returns of each state meet our expectations.

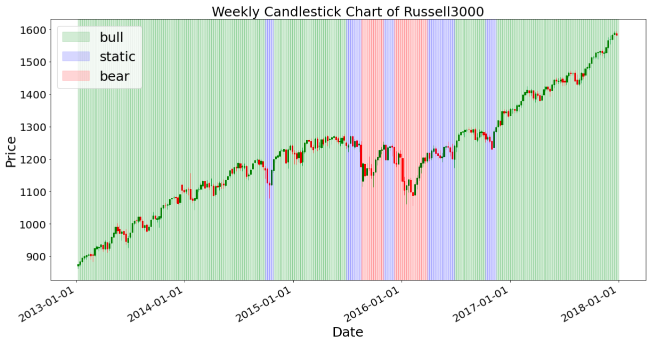

Fig. 15: Candlestick Chart with Detected Regimes of Validation

More importantly, the t-test table shows that the volatility means between bull and bear states, as well as static and bear states, are now significantly different from each other at a 95% confidence level. Taking a further look at the standardized volatility means, we find that the clustered bear market has the highest volatility among the three states while the bull market has the lowest. This conclusion also aligns with the result of the three-state variance switching model between 2013 and 2018.

Bonferroni corrected α = 0.016	t-test		
Green cells means the pair is different			
p-value	Bull	Static	Bear
Bull		0.1228	4.24E-09
Static	0.1228		0.0114
Bear	4.24E-09	0.0114	
Mean and std of volatility	Bull	Static	Bear
Mean	-0.78	-0.69	-0.41
Std	0.31	0.51	0.24
Mean and std of return	Bull	Static	Bear
Mean	0.003	0.000	-0.001
Std	0.012	0.019	0.026

Table 6: T-test Table of Testing

Figure 16 displays the market regime predictions by LSTM before and after we add the two features. The new prediction aligns with our expectation of a timelier reaction to the regime switch after the COVID 19 market crash. In addition, our LSTM model achieves a recall ratio of 88.89% for the bear states, 89.66% for the static states, and 96.50% for the bull states. The overall accuracy score is 94.45%. Therefore, we conclude LSTM remains effective under the updated regimes.

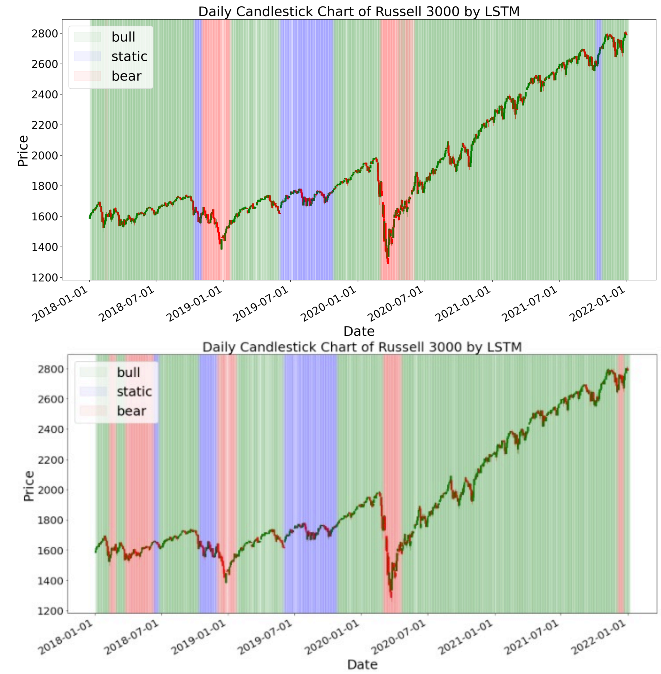

Fig. 16: Comparison of Daily Candlesticks Predicted by LSTM

B. Updated Determination Backtest

We perform the backtesting again to confirm the improvement of our newly determined regimes. Figure 17 and Table 7 indicate that the switching strategy based on the improved regimes can realize a higher cumulative return with better downside protection. This is mainly because the newly determined states alleviate the problem of response lag, allowing our switching strategy to respond more promptly to the point of the regime turnaround.

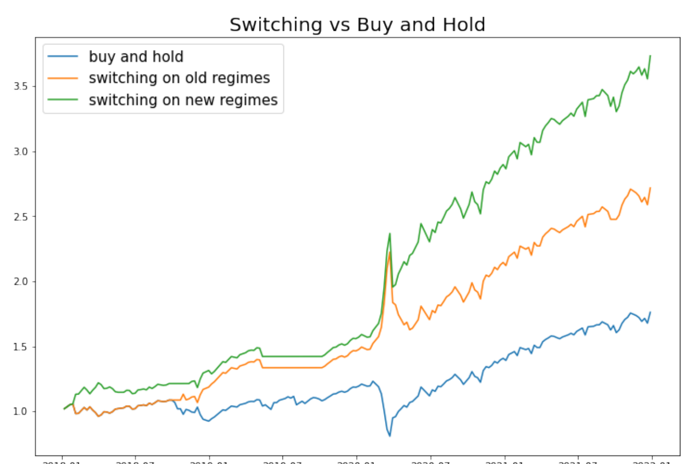

Fig. 17: Cumulative Return of Switching Strategy on Determined Regimes vs. Buy-and-Hold Strategy

Strategy		Cumulative Return	Sharpe ratio	Omega ratio	Max drawdown	Calmar ratio
Buy-and-Hold		1.762	0.77	1.17	-35.09%	0.44
Switching	Old	2.763	1.35	1.35	-26.74%	1.11
	New	4.255	1.90	1.54	-16.22%	2.77

Table 7: Performance of Switching Strategy on Determined Regimes vs. Buy-and-Hold Strategy

C. Updated Prediction Backtest

Similarly, we backtest our regime prediction. Both Figure 18 and Table 8 illustrate that, after improving our original clustering, the switching strategy on top of our predicted states also accomplishes a comparable boost in risk-adjusted returns and can now outperform the buy-and-hold strategy.

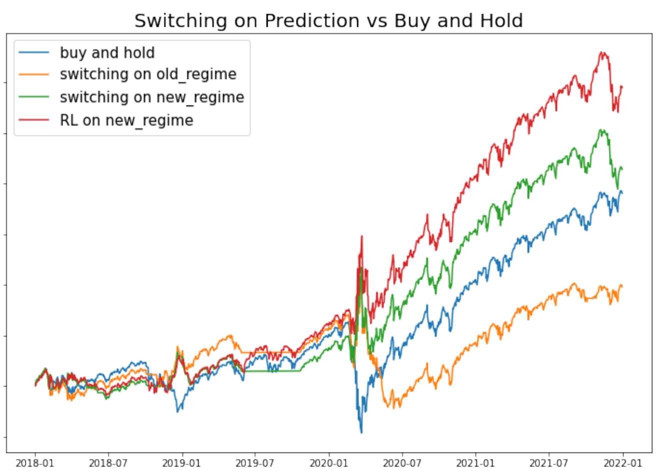

Fig. 18: Cumulative Return of Switching Strategy on Predicted Regimes vs. Buy-and-Hold Strategy

Strategy		Cumulative Return	Sharpe ratio	Omega ratio	Max drawdown	Calmar ratio
Buy-and-Hold		1.762	0.77	1.17	-35.09%	0.44
Switching	Old	1.392	0.50	1.12	-36.77%	0.24
	New	1.855	0.86	1.21	-23.38%	0.73
RL_LSTM		2.180	1.10	1.24	-21.70%	1.03

Table 8: Performance of Switching Strategy on Predicted Regimes vs. Buy-and-Hold Strategy

VIII. REINFORCEMENT LEARNING

The switching strategy requires us to either go long, short selling or invest in the bond market with all of our assets, which is more or less unrealistic. Now that we have a proper market state prediction, we would like to replicate a human trader's behavior as much as possible and evaluate if our definition of market states can boost the trading performance. Reinforcement learning comes into play as we can train an agent to learn the optimal policy for the portfolio allocation in connection with the dynamically changing market conditions.

The actor-critic algorithm consists of two networks (the actor and the critic) to solve our problem. The Advantage Function calculates the agent's TD Error or Prediction Error at a high level. The actor-network adjusts the portfolio at each step, and the critic network evaluates the quality or the Q-value of a given input state. As the critic network learns which states are better or worse, the actor uses this information to teach the agent to seek out good states and avoid bad states [6].

In our case, we create three agents and train them separately under the predefined regimes. According to our prediction of the market states, we then pick the right agent to manage the portfolio. A set of most common technical indicators such as RSI, CCI, and KDJ are provided to help the agents learn the optimal policy. We also train an agent without any information about the market regimes to compare the performance.

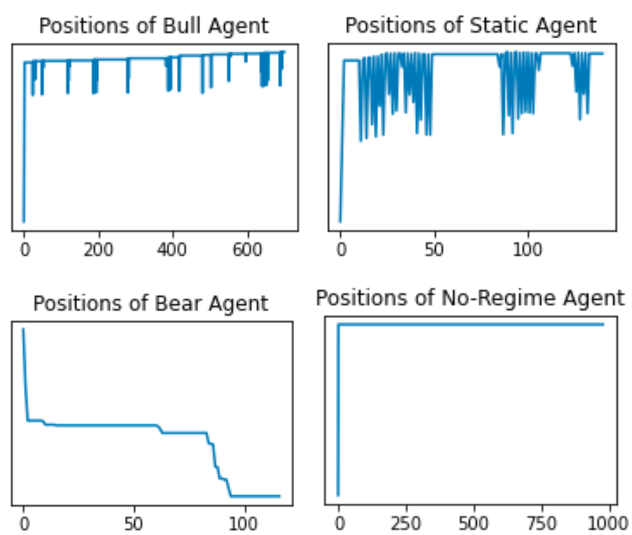

Fig. 19: Holding Positions of Reinforcement Agents

Based on Figure 18 and Table 8, our reinforcement learning strategy has the highest cumulative returns and Sharpe ratios, as well as the lowest maximum drawdown among all of our strategies. Moreover, the backtesting result looks promising intuitively since the bull market agent buys and holds the market most of the time and occasionally reduces the position when the market is overbought. On the other hand, the bear market agent shorts the market as much as it can. The static market agent relies heavily on technical indicators and trade the market most frequently. Furthermore, the agent without any regime context finds its optimal action is to buy and hold the market. The conclusion is inspiring as it suggests that the optimal strategy for most investors is to buy and hold the market.

IV. CONCLUSION

In this paper, we compare different clustering models and find HMM to be the most effective to determine the hidden market states: bull, bear, and static. Then we compare XGBoost and LSTM for a reliable prediction of the near-future regime. LSTM stands out to be a better prediction model but has the problem of lagged reaction to regime reversals.

To solve this problem, we look for new features to feed into HMM and hope a better regime determination can lead to a better prediction. After adding the Percentage of Russell 3000 stocks with price above the 50-day moving average and VIX index, our switching strategy is able to outperform the buy-and-hold strategy based on the predicted market states. Finally, we develop a more reasonable and realistic trading strategy with reinforcement learning. The combination of our improved market regimes and advanced trading strategy enables us to obtain the highest risk-adjusted returns among all strategies.

In the long term, we can use natural language processing on unstructured data like the latest news or twitters to generate meaningful features on market sentiments. We believe that, by integrating sentimental analysis into our clustering and predicting models, we can catch the significant trend reversal points like the COVID 19 in a timelier fashion.

REFERENCES

[1] Ace B. (2021) Stock Market Indicators [What They Are, Types & Benefits]. https://review42.com/resources/stock-market-indicators

[2] Bernard R., Manuel A., Florent D., Pascal V., Stéphane B., et al. (2021) Methods to choose the best Hidden Markov Model topology for improving maintenance policy. MOSIM'12 9th International Conference of Modeling, Optimization and Simulation, Jun 2012, Bordeaux, France.

[3] Botte, A., & Bao, D. (2021). A Machine Learning Approach to Regime Modeling. Two Sigma. http://www.twosigma.com/articles/a-machine-learning-approach-to-regime-modeling/

[4] Chang-Jin. K., Charles. R. N., Richard S. (1998) Testing for mean reversion in heteroskedastic data based on Gibbs-sampling-augmented randomization. Journal of Empirical Finance 5 131–154.

[5] He, X., D. Cai, and P. Niyogi. (2005) Laplacian Score for Feature Selection. NIPS Proceedings.

[6] Mike W. (2021) Advantage Actor Critic Tutorial: minA2C. https://towardsdatascience.com/advantage-actor-critic-tutorial-mina2c-7a3249962fc8

[7] Saxena S. (2021) LSTM: Introduction to LSTM: Long Short Term Memor. Analytics Vidhya, https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/