Extraction of the Essential Constituents of the S&P 500 Index

Ray R. Hashemi
Department of Computer Science
Armstrong State University
Savannah, GA, USA
rayhashemi@gmail.com

Omid M. Ardakani
Department of Economics
Armstrong State University
Savannah, GA, USA
Omid.Ardakani@armstrong.edu

Azita A Bahrami
IT Consultation
Savannah, GA, USA
Azita.G.Bahrami@gmail.com

Jeffrey A. Young
Department of Computer Science
Armstrong State University
Savannah, GA, USA
alanyoung@gmail.com

Abstract— The S&P 500 index is a leading indicator of the stock market and U.S. equities which is highly influenced by its essential constituents. Traditionally, such constituents are identified by the market capitalization weighting scheme. However, the literature rejects the efficiency of the weighting method. In contrast, we introduce data mining approaches of the entropy and rough sets as two separate methods for extraction of the essential S&P 500 constituents. The legitimacy of the findings in comparison with the S&P 500 weighting scheme have been investigated using the discrete time Markov Chain Models (MCM) and Hidden Markov Chain Models (HMCM) which lend themselves easily to the nature of the time-series data. The investigation is done against data for the full sample and pre/post crisis subsamples collected for the period of 16 years. We find the entropy method provides the highest forecasting accuracy measure for the full sample and post-crisis subsample.

Keywords—S&P 500 Index, Essential Constituents of the SP index, Entropy, Rough Sets, Markov Chain Model, Hidden Markov Chain Model

I. INTRODUCTION

To study the performance of the stock market, investigators consider different indices such as the Standard & Poor's 500 index (hereafter SP). The SP is a leading indicator of the stock market and U.S. equities. It was introduced in 1957 to track the trend of stock prices for 500 large corporations in the United States. The companies are listed as the SP constituent if they meet the following preconditions: (a) the market capitalization must be greater than or equal to $6.1 billion (the market cap is defined as the share price times the number of shares outstanding), (b) the ratio of annual dollar value traded to cap should be at least 1.0, and (c) minimum monthly trading volume must be 250,000 shares [1]. The SP for a given constituent may be calculated using the log returns defined by $r_t = \log(p_t/p_{t-1})$, where p_t is the adjusted stock price for the constituent on day t. The SP movements are affected by many factors such as macroeconomic indicators, political events, and world news [2][3]. When it comes to calculation of the SP some of these constituents are more influential than the others and we refer to them as the *essential constituents*. The top 10 companies reported in the literature [3][4][5] are considered as the essential constituents of the SP.

The interpretation of the influence of an essential constituent over the SP is that it mimics the behavior of the constituent moves. In other word, a constituent may be used as a predictor of the SP. Higher the influence of a constituent means it is a more accurate predictor of the SP. The major question we answer in this paper is whether there are essential constituents of the SP other than the top companies reported in literature and if so then, how essential they are in compare to the reported 10 companies. The goal of this research effort is twofold: (a) use of data mining to extract the other lists of essential constituents, if any, from the time series of the daily returns indices reported for all the constituents of the SP and for the last 16 years and (b) establish the legitimacy of the extracted lists by investigating and comparing every list's power of forecasting the SP with the forecasting power of the top 10 list of Standard & Poor's.

The rest of the paper is organized as follows. The previous works is the subject of Section 2. The methodology is presented in Section 3. The empirical results are discussed in Section 4. The conclusions and future research are covered in Section 5.

II. PREVIOUS WORKS

In the literature various methodologies are used to forecast the SP but little is done to identify its essential constituents. For example, in reference to forecasting, Heaton et al. use deep learning hierarchical models to approximate the SP based on the top ten companies in the index. These companies are selected using the Standard and Poor's weighting scheme [5]. By doing so, they extract non-linear features from the constituents. Krauss et al. find the effectiveness of three methods for the SP forecasting [6]. They use the returns of all stocks in the SP companies from 1992 to 2015 on a daily basis and compare deep neural network, gradient boosted-trees, and random forests in the context of statistical arbitrage. They find random forests outperform the other two methods. Yudong and Lenan propose the bacterial chemotaxis optimization in the context of back propagation neural network for forecasting the stock market indices [7]. Vargas et al use a combination of top companies' stock prices, macroeconomic indicators, and market sentiments to predict the SP fluctuations [8][9].

In reference to identification of the essential constituents of the SP, Standard and Poor's ranks the essential constituents using the market capitalization weighting scheme. However, the literature rejects the efficiency of capitalization-weighted indices [10][11]. To the best of our knowledge, the literature

978-1-5386-2652-8/17 $31.00 © 2017 IEEE
DOI 10.1109/CSCI.2017.59

lacks a comprehensive study that extracts the essential constituents and validates their legitimacy. To fill such a void, we not only introduce a methodology for extracting the essential SP constituents but we also present the methodology for establishing the legitimacy of the extracted constituents.

III. METHODOLOGY

Let the time series dataset of TD be the one that contains the daily stock returns along with the SP for the last Y years (TD has 252(days/year)*Y records.) The methodology that will be shortly introduced provides for preprocessing of the TD dataset, extracting the essential constituent lists from TD using different datamining approaches, and examining the legitimacy of the extracted lists. The details are covered in the following three subsections.

A. Preprocessing

The preprocessing of the TD dataset is necessary to prepare it for data mining. Such preparation is completed in the following three sub-steps: Imputing the Missing Data, Testing of Structural Change, and Discretization of Data.

1) Imputing the Missing Data: We use the Fully Conditional Specification (FCS), also known as Multivariate Imputation by Chained Equations (MICE) [12] to impute the missing data in the TD dataset. The MICE algorithm preserves the relationship among the observed vector and imputes the missing values. This method assumes the missing data are in random. Let A_i for i = 1, . . . , p be an incomplete variable including observed, A_i^{obs}, and missing, A_i^{mis} parts. The h-th imputed data is derived by the chained equations. The complete data including observables and imputed values come from a multivariate distribution P(i|θ), where θ is a vector of unknown parameters. The MICE algorithm obtains the posterior distribution of θ by sampling iteratively from conditional distributions. The iteration of chained equations is a Gibbs sampler that draws from the posterior distribution of θ given A^{obs} and previous iteration of A_i.

2) Structural Change: To consider the structural change in the data due to external shocks such as the financial crisis we estimate breaks in our time series model. We implement the algorithm introduced by Bai and Perron [13] to find breakpoints in the data. Their method estimates the breakpoints by minimizing the residual sum of squares (RSS) from the unconditional mean model in which the dependent variable is the SP and the independent variable is the intercept. The RSS computes the residual sum of squares for a segment starting at observation i ending at j, where i<j.

3) Discretization of Dataset: Each attribute, A_i, of the dataset is discretized separately using the following clustering technique. The range of [Max - Min] values in A_i are divided into Γ random size partitions, where Γ is rather a large number close to $|A_i|$. One seed from each partition is selected (generally, the median value of the partition is chosen.) The clustering technique makes each value of A_i a member of a cluster for which the value and the seed of the cluster have the smallest Euclidean distance. The seed of the cluster is then replaced by the average of the value and the seed. Ultimately, the clustering technique delivers Γ clusters for A_i. All the values in a given cluster are replaced by one discrete value. It is necessary to provide resolutions for the cases when one or more clusters are extremely small or extremely large.

If one or more clusters are extremely small then, we can use either a *total* or *partial* resolution. The total resolution dismisses the current seeds, decrements the number of partitions (Γ) by one and then repeats the entire process of seed finding and cluster generation. The partial resolution (a) merges the underlying partition for the small cluster with one of its adjacent partitions (the one that its corresponding cluster is smaller and in the case of having a tie, one is chosen randomly), (b) selects a new seed for the merged partition, and (c) repeats the clustering process.

If one or more clusters are extremely large then, we can again use either a *total* or *partial* resolution. The total resolution undoes the clustering, increases the number of partitions by one and then the process of clustering is repeated. The partial resolution further divides the underlying partition of the large cluster to increase the number of seeds by one for that specific partition and then the clustering process is repeated only for that partition. In practice, we start with a large value for Γ and then by the process of trial and error the right value for Γ is identified.

B. Extracting the Essential Constituents

The preprocessing step imputed the missing data from the time series dataset of TD, found the breakpoint, and discretized the dataset. The TD dataset has the S&P 500 companies as the independent variables and the SP as the dependent variable. We try to determine the level of influence that each independent variable has over the SP and then Q independent variables with the highest level of influence are picked as the essential constituents. To do so, we use two different approaches of the Entropy and Rough Sets.

1) The Entropy approach that has its root in information theory and it determines the influence level of each independent variable on the dependent variable by the amount of information gain of the independent variable in reference to the dependent variable [14]. Therefore, the independent variable with the highest gain has the highest level of influence over the dependent variable. To describe the entropy approach further, let attributes (A_1 . . . A_n) and attribute D be the independent variables and dependent variable of the TD dataset, respectively. Let the unique values (classes) in D be d_1 . . . d_g. The entropy of TD is defined by formula (1):

$$ENT(TD) = -\sum_{i=1}^{g} p_i * log_2 p_i, \quad (1)$$

Where p_i is the probability of class k_i in D. Let the unique values for attribute A_j be (v_1 . . . v_n). Let also V_i be the dataset made up of those records of TD for which the value of A_j= v_i. The value for $H(A_j)$ is defined by formula (2)

$$H(A_j) = \sum_{i=1}^{n} \omega_i * ENT(V_i), \quad (2)$$

where $\omega_i = |V_i|/|TD|$. The information gain for A_j is defined by formula (3):

$$Gain(A_j) = ENT(TD) - H(A_j). \quad (3)$$

2) The Rough Sets approach was introduced by Pawlack in 1984 [15]. The details may be found in [16][17]. In contrast with the traditional set, rough set does not have a crisp boundary but has a boundary region which means its members are inside of the set, outside of the set, or on the boundary region of the set. Therefore, a rough set is defined by its lower and upper approximation spaces. Let U be a universe of objects (a non-empty and finite set) and let $E \subseteq U$. The lower and upper approximation spaces for the rough set, E, are defined by formulas (4) and (5):

$$\text{Lower}(E) = \{a \in U \mid [a]_{\Re} \subseteq E] \quad (4)$$

$$\text{Upper}(E) = \{a \in U \mid [a]_{\Re} \cap E \neq \emptyset], \quad (5)$$

where $[a]_{\Re}$ is a binary equivalence relation over U. The boundary region of the rough set E is defined by formula (6):

$$\text{Boundary}(E) = \text{Upper}(E) - \text{Lower}(E) \quad (6)$$

Let B be the set of all equivalence classes in dataset TD which represents the internal structure of the TD. All the objects in one equivalence class are duplicates of each other. The major task is to reduce the attributes of TD without affecting the members of the set B. In nomenclature of rough sets, a minimal subset of the TD attributes that can preserve B is called a reduct. We consider the companies in reduct of TD as the essential constituents of TD. The reduct is generated by iterating the following two steps:

a. Flagging every attribute in TD that its removal, in presence of the rest of attributes, does not modify B.
b. Selecting randomly one of the flagged attributes for permanent removal from TD.

TD may have several reducts and the reason stems from the fact that when more than one attribute is eligible for removal and randomly selecting one to be removed has a major effect on the next iterations. The reduct with the smallest number of attributes is the reduct of interest. Finding such a reduct demands creation of all possible reducts for TD.

Let Δ be the reduct of interest with μ attributes. Let also Q be the maximum number of essential constituents that one has in mind to extract by use of the rough sets approach. If $\mu \leq Q$ then, there is no need for further action to take and $Q = \mu$. However, if $\mu > Q$ then the following selection process takes place.

All subsets of Δ are created ($\delta_1 \ldots \delta_f$) such that each subset is made up of Q independent attributes plus the dependent variable. The boundary region for each subset is calculated using formulas (6) The subset δ_i is considered the winner for which the $|\text{Boundary}(\delta_i)| = \text{Min}(|\text{Boundary}(\delta_i)|, \ldots |\text{Boundary}(\delta_f)|)$. In the case of tie one chosen randomly.

C. Formal Method for the Legitimacy Examination of the Extracted Lists

Let the separate application of two different methodologies on the TD dataset delivers two subsets of the attributes of TD (K_1 and K_2) as two lists of the essential constituents for the SP such that: (a) $K_1 \cap K_2$ may or may not be empty and (b) $|K_1| = |K_2| = Q$. Let W_1 and W_2 also be two projections of the TD dataset over K_1 and K_2. The SP as the dependent variable is added to both W_1 and W_2. The formal methodology that is presented in this section will determine which one of the two subsets of K_1 and K_2 is more influential over SP and why. The premise of our argument is that since K_1 and K_2 sets of companies are greatly influence the SP then, the two sets can serve as two different predictors of the future SP. The one with a higher accuracy of the prediction includes companies that have a higher influence on the SP and, therefore, it represents more accurate list of top Q companies in TD.

To make K_1 and K_2 as two different predictors of the SP we use both Markov Chain Models (MCM) and Hidden Markov Model (HMM). To do so, first we present the MCM and HMM in the following two subsections and then our formal method of legitimacy examination is presented as the last subsection.

1) Markov Chain Model: A discrete-time Markov chain is a directed graph that can model behavior of a stochastic system with the following properties:

a. System has a set of N states, $S = (s_1 \ldots s_n)$ and a set of probability of transitions from each state to all the other states in S,
b. System is resided in one of the states of S for each discrete period of time, and
c. Prediction of the future state in which the system resides is only dependent on the current state that system is in and not the past states of the system.

A Markov matrix, M, is built out of the probability of transitions such that it has N rows and N columns:

$$M = \begin{array}{c} \\ s_1 \\ \cdot \\ \cdot \\ \cdot \\ s_n \end{array} \begin{array}{c} \begin{array}{cccc} s_1 & . & . & . \end{array} \; s_n \\ \begin{bmatrix} p_{11} & . & . & . & p_{1n} \\ & & \cdot & & \\ & & \cdot & & \\ & & \cdot & & \\ p_{n1} & . & . & . & p_{nn} \end{bmatrix} \end{array}$$

The j-th row of the matrix, $\text{row}_j = (p_{j1}, \ldots p_{jn})$, is a probability vector represents the probability of transition from s_j to other states in S including itself. The probability of p_{ji} is calculated by Maximum Likelihood Estimation using formula (7):

$$p_{ij} = P(s_j \mid s_i) = \frac{Number\ of\ observed\ transitions\ from\ s_j\ to\ s_i}{Total\ number\ of\ observed\ transitions\ from\ s_j} \quad (7)$$

Let the current time period be t_0, and the current state of the system be presented by a probability vector, V_0, with the length of N such that the element$_j$ of the vector determines the approximated probability of the system being in the state of s_j and it is calculated using formula (8):

$$P_{element\ j} = \frac{Frequency\ of\ observed\ s_j}{Total\ number\ of\ observed\ states} \quad (8)$$

At time period of t_1 and t_2, the current state of the system could be presented by the probability vectors of V_1 and V_2 where, $V_1 = V_0 * M$ and $V_2 = V_1 * M = V_0 * M^2$. Using the same analogy, the probability vector after k period of time is given by formula (9):

$$V_k = V_0 * M^k \quad (9)$$

The state corresponding to the highest probability in vector V_k is the state in which the system highly likely to reside after k time periods. If the value of k grows beyond a point then, the content of the matrix M will not change. The minimum value of k for which the matrix M^k no longer changes is referred to as the *ceiling of k, k_{ceil}* in this paper.

2) Hidden Markov Chain Model: The Hidden Markov Chain Model (HMM) represents a sextuple system of (s_0, S, O, V, M, M_{os}) Where,

- s_0 is a starting state,
- S is a set of N hidden states, S = {s_1 . . . s_n},
- O is a set of F observed states, O = {o_1 . . . o_f},
- V_0 is a probability vector for the current state of the system,
- M is the transition probability matrix of for the hidden states, and
- M_{os} is the emission probability matrix from hidden states to observed states.

The probability vector V_0 and the transition matrix M are the same as V_0 and M defined for MCM. Each element of matrix M_{os}, m_{ji}, = $P(o_i \mid s_j)$. For the HMM of Figure 1, S = {s_1, s_2, s_3}, O = {a, b, c}, V = (0.6, 0.1, 0.3),

$$M = \begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \end{array} \begin{array}{c} \begin{array}{ccc} s_1 & s_2 & s_3 \end{array} \\ \begin{bmatrix} 0.2 & 0.7 & 0.1 \\ 0.1 & 0.3 & 0.6 \\ 0.3 & 0.5 & 0.2 \end{bmatrix} \end{array}, \text{ and } \quad M_{os} = \begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \end{array} \begin{array}{c} \begin{array}{ccc} o_1 & o_2 & o_3 \end{array} \\ \begin{bmatrix} 0.4 & 0.2 & 0.4 \\ 0.3 & 0.6 & 0.1 \\ 0.2 & 0.2 & 0.6 \end{bmatrix} \end{array}$$

The model is used to determine the best sequence of hidden states and emission probabilities for a given output sequence. Let us consider the HMM of Figure 1and output string of ST = "abbc". The move from s_0 is an epsilon move, therefore, our ST = "ɛabbc". We make a change in Figure 1 to help us finding the possible paths easier by absorbing the emission probabilities into transition probabilities, Figure 2. Each transition arch among the hidden states is annotated by all observed states and their probabilities and it is calculated simply by the $M_{ij} * P(o_i \mid s_i)$. The reason for not absorbing the emission probabilities into probability vector V_0 is the observed state for s_0 is ε.

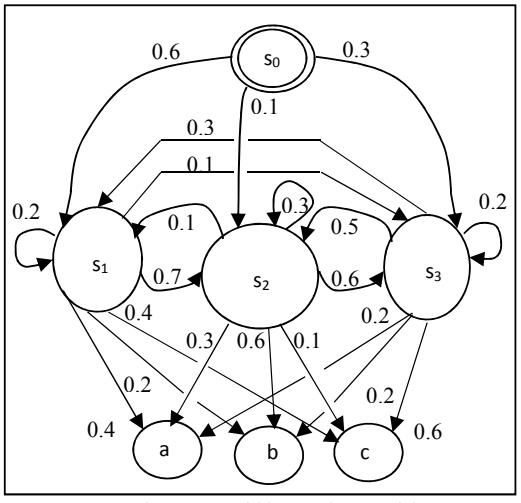

Figure 1: A Hidden Markov Model

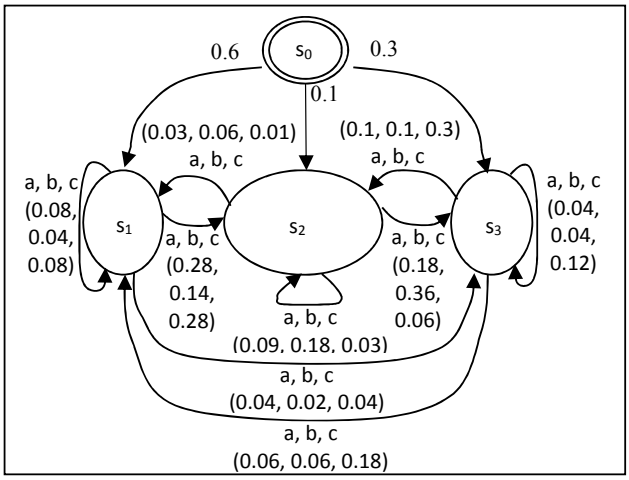

Fig. 2: Absorption of emission probabilities into transition probabilities of hidden states

The number of possible paths = $(N)^F = 3^3 = 27$. However, we are interested in a sequence of hidden states that delivers a path with the highest probability. This makes the process of finding the sequenced states computationally expensive even for small number of hidden states and observed states. The well-known method for finding the optimal path among all the possible choices for a given sequence is the Viterbi algorithm that uses dynamic programming [18]. The details of this algorithm may be found in [19][20][21]. In a nutshell, we build a tree structure for which the root is s_0. The number of branches from s_0 is equal to N (number of hidden states) and to each branch a probability is assigned using the probability vector V_0. Each hidden state makes a new node of the tree which in turn has N branches and N new leaves. Each branch has an initiating state and an ending state.

The probability assigned to each branch in the second expansion (and for the rest of expansions) is the *absorbed* emission probability of the next symbol in ST (observed state) for moving from initiating state to the ending state of the branch using Figure 2. Each node of the tree is also assigned a probability which is the product of the probability of each branch in the path that starts from the root and ends at the node. The tree grows to L levels where L = Length(ST).

Traversing the tree to find the optimal path is an expensive process. To minimize the cost we apply a filtering technique (Viterbi Algorithm) at each level of the tree such that at the level l only one leaf per hidden state will survive and that is the one with the maximum probability. Therefore, at each level we have exactly N survived leaves. Only the survived nodes will be expanded and the rest of the nodes are ignored. After tree goes to L levels only N leaves are survived among which one of the leaves has the highest node probability and that is the final hidden state for the system in reference to ST. Through the filtering process, "tie" cases may happen in selecting a node for expansion. Such cases are resolved by randomly chosen one of the nodes among the tie ones.

3) The Formal Method: Let the dataset W_1 have N records of $R_1 \ldots R_n$ such that R_1 includes returns of the K_1 companies along with the SP for *today*. R_2 includes the Return indices and the SP for *yesterday*, and R_n includes the same type of data for the *n-th day in the past* starting from today. We divide W_1 into a pair of training and test sets by choosing the first consecutive G records ($G << N$) starting from R_1 and ending at R_g, ($R_1 \ldots R_g$), as the test set and the remaining records in W_1 as the training set ($R_{g+1} \ldots R_n$). We train K_1 Markov Chain Models (MCM_i, for i = 1 to $|K_1|$, one per company appeared in K_1.

The set of states for MCM_i that models the company C_i of K_1 represents the set of the unique Return values for C_i. The probabilities of transitions from each state are calculated by applying formula (7) on the data for C_i in the training set which leads to building of the Markov matrix, M_i. The MCM_i is trained when the building of its M_i is completed.

Considering the training set, let the probability vector for C_i be V_i at time period of t_0 (current time, which this is the time period belong to the record R_{g+1}) for the k-th time period in future, the predicted returns value for C_i is the value represented by state s_j and it is determined by formula (10):

$$s_j = \text{argmax}(V_i * M_i^k). \quad (10)$$

The predicted return value is compared with the return value for C_i by record R_{g-k} of the test set which is the actual return value for C_i in k time period in future from the record R_{g+1} of the training set. One can predict and compare the return value for C_i using k = 1 to g-1 to establish a measure of accuracy, α_i, for MCM_i prediction performance which is simply the percentage of the correct classification of the future C_i return values. The *overall prediction certainty factor (MCM_{cf})* for prediction of return values using MCMs is calculated by *weighted-α*. To do so, a weight is given to each company in K_1 which is equal to |Q|–LOC+1, where LOC is the order number of the company in K_1. Recall the company at the beginning of the list has the highest influence over the SP. Thus, the MCM_{cf} is calculated using formula (11):

$$MCM_{cf} = \frac{\sum_{i=1}^{|Q|} \alpha_i . w_i}{\sum_{i=1}^{|Q|} w_i}. \quad (11)$$

Let M_i be the Markov matrix of MCM_i trained by the training set of company C_i and let G be the size of its test set. It is important that $G \leq K_{ceil}$ of M_i be true; Otherwise, the test records of $R_{g-kceil}$ cannot be used in the testing process. Therefore, k_{ceil} of M_i dictates the value of G.

We use the K_1 Markov Chain Models to build a record of the Return values for the given time period of k = 1to g in future. Each built record, R_h, in turn, is fed as input to the Hidden Markov Model that is built out of the training set of W_1 to predict the SP for R_h. The accuracy of predicted SP is checked against the SP of the record R_{g-h} of the test set. The percentage of correct classification used as a measure of accuracy, β, for HMM prediction performance. The essential constituents S&P500 companies has the *overall prediction certainty factor* of $\gamma = MCM_{cf} * \beta$.

The same process is repeated for W_2 by building K_2 new Markov Chain Models and one new Hidden Markov Chain model using the training set of W_2. And calculate the certainty factor for overall prediction of the essential constituents of the SP, $\gamma' = MCM'_{cf} * \beta'$. The higher overall prediction certainty factor between W_1 and W_2 identifies the one with more legitimate set of essential constituents of the SP. Such claim stems from the fact that the subset with higher γ is more influential on the SP than the other set.

IV. Empirical Results

The imputed and discretized dataset of the 16 years (2001 to 2017) of the daily stock returns made the dataset of *Full* with 4075 records and 504 attributes (503 companies and the SP.) The discrete data values were three for every attribute which delivered by the clustering approach. The investigation of the structural change in the Full dataset reveals only one break on March 12, 2009 where its associated RSS is 0.612. Thus, we split the sample into pre and post crisis subsamples. To summarize, based on the structural test results, we define the following datasets: (1) Full: January 8, 2001 to March 21, 2017 with 4075 records, (2) Pre: January 8, 2001 to March 11, 2009 with 2054 records and (3) Post: March 12, 2009 to March 21, 2017 with 2021 records.

Two essential constituent lists (with the length of ten) were extracted from each of the three datasets of Full, Pre, and Post using two different extraction approaches of entropy and rough sets (total of six lists). The projection of Full, Pre, and Post datasets over the six lists delivered six new datasets and they were named using the dataset names of Full, Pre, and Post subscribed by the approach names of the entropy (e) and rough sets (r). In addition, the Full, Pre, and Post datasets were projected over the list of top ten companies delivered by the Standard & Poor's approach (p) resulting in three new datasets that were named following the same annotation used for the other six datasets. The SP was also added to each one of the new nine datasets. As a result, the following nine datasets (that collectively make the *working-set*) were produced: Pre_e, Pre_r, Pre_p, $Post_e$, $Post_r$, $Post_p$, $Full_e$, $Full_r$, and $Full_p$. Each member of the working-set has eleven attributes (ten essential constituents + the SP) and the list of the essential constituents for each member is shown in Table 1.

For each member of the working set, a pair of training and test sets was identified. The size of the test set for all members of the working set was 150 records (5 month worth of data) determined by the minimum k_{ceil} among all members of the working set. For each member also ten Markov Chain Models were built to predict the Return values for each company. The *overall prediction certainty factor (MCM_{cf})* for each company was calculated using formula (11) and displayed in Table 1.

Each member of the working set used its own ten MCMs to build a record for k time period in future (k = 1 to 150.) As a result, each member created its own dataset of 150 future records by using its own MCMs. One Hidden Markov Model was built per member to predict the SP for each one of the 150 future records of the member. The accuracy of such predictions (β) for each member is calculated and shown in Table 1. The *overall prediction certainty factor* of, γ, for each member is also calculated and displayed in Table 1.

TABLE 1: THE WORKING SET, ITS MEMBERS, AND THEIR RESULTS OF ANALYSIS

	WORKING SET MEMBERS								
	Pre_e	**Pre_r**	**Pre_p**	**$Post_e$**	**$Post_r$**	**$Post_p$**	**$Full_e$**	**$Full_r$**	**$Full_p$**
T	FOXA	YHOO	GE	ITW	VMC	AAPL	AME	WFM	AAPL
o	CVX	XL	XOM	APD	TRIP	MSFT	JEC	YHOO	MSFT
p	EQT	WFM	MSFT	UNP	HSY	AMZN	RHI	VFC	FB
	ITW	ZION	T	PCAR	UAL	FB	SPGI	VIAB	JNJ
T	RHI	VRSK	C	UTX	VRTX	JNJ	PH	ZION	GE
e	CMI	WMB	PG	ETN	WDC	XOM	ITW	VMC	JPM
n	PH	VFC	WMT	PPG	VFC	JPM	PCAR	WMB	WFC
	SPLS	ZTS	JNJ	PH	VTR	GOOGL	XRAY	VTR	PG
C	JEC	UHS	AIG	DHR	TSS	GOOG	SNA	TSS	T
o	PCAR	YUM	WFC	FISV	SYF	WFC	BEN	YUM	WMT
MCM_{cf}(%)	**0.65**	**0.63**	**0.71**	**0.89**	**0.61**	**.84**	**0.83**	**0.47**	**0.78**
β(%)	**0.71**	**0.64**	**0.75**	**0.96**	**0.75**	**0.89**	**0.91**	**0.75**	**0.83**
γ(%)	**0.46**	**0.40**	**0.53**	**0.85**	**0.46**	**0.75**	**0.76**	**0.35**	**0.65**

V. CONCLUSION AND FUTURE RESEARCH

The results show that rough set and entropy approaches provide different sets of companies and both differ from the Standard and Poor's publication which ranks the companies based on a weighting scheme. For the full sample (Full), the overall prediction certainty factor is the highest for the entropy method and lowest for the rough set approach. The findings illustrate that the entropy method is the most accurate in terms of forecasting the SP among the three. This holds true for the subsample of the post-crisis period (Post). However, we find different results for the pre-crisis subsample (Pre) in which the weighting scheme outperforms the entropy and rough set methods. Intuitively, the weighting scheme might work for the period before the financial crisis, but the method has inferior performance for the post-crisis and full samples. This finding signifies the importance of examining the SP movements by looking at different methodologies. Overall, the results show that shares of stocks with higher market cap do not necessarily explain the SP fluctuations. This study may have implications for both researchers and practitioners, because it introduces different methods for extracting the essential constituents.

As future research, the effect of extracted essential constituents on the traditional forecasting methodologies is under investigation. We also plan to expand the capital asset pricing model using the proposed extraction methodology.

REFERENCES

[1] S&P U.S. Indices Methodology, Standard & Poor's. 2017.

[2] E.F. Fama, *Foundations of Finance*, New York, 1976, pp. 133–167

[3] D. M. Cutler, J. M. Poterba, and L. H. Summers, "What moves stock prices?", The Journal of Portfolio Management, 1989, 15, pp.4–12.

[4] J. J. Siegel and J. D. Schwartz, "Long-term Returns on the original S&P 500 companies". Financial Analysts Journal, 2006. 62(1), pp.18–31.

[5] J. B. Heaton, N. G. Polson, and J. H. Witte, "Deep learning in finance: deep portfolios", Applied Stochastic Models in Business and Industry, 2009, 33(1), pp.3–12.

[6] C. Krauss, X. A. Do, and N. Huck, "Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500", European Journal of Operational Research, 2017, 259(2), pp.689–702.

[7] Z. Yudong and W. Lenan, "Stock market prediction of S&P 500 via combination of improved BCO approach and BP neural network", Expert Systems with Applications, 2009, 36(5), pp.8849–8854.

[8] M. R. Vargas, B. S. L. P de Lima, and A. G. Evsukoff, "Deep learning for stock market prediction from financial news articles", 2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA) : 2017 Pages: 60 – 65

[9] A. Joseph, M. Larrain, C. Turner, "Daily Stock Returns Characteristics and Forecastability", Procedia Computer Science Volume 114, 2017, Pages 481-490.

[10] R. D. Arnott, J. Hsu, P. Moore, "Fundamental Indexation", Financial Analysts Journal, 2005, Vol 61, Issue 2: 64-85

[11] P. D. Kaplan, "Why Fundamental Indexation Might—or Might Not—Work", Finanacial Analysts Journal, 2008, Vol. 64, No. 1: 32-39.

[12] K. Groothuis-Oudshoorn, and S. Van Buuren, "mice: Multivariate Imputation by Chained Equations in R", Journal of Statistical Software, 2011, 45(3), pp.1–67.

[13] J. Bai and P. Perron, "Computation and analysis of multiple structural change models", Journal of Applied Econometrics, 2003, 18(1), pp.1-22.

[14] J. R. Quinlan, "Learning Efficient Classification Procedures and Their Application to Chess Endgames. In: J. S. Michalski, J. G. Carbonell, T. M. Mirchell, Editors, Machine Learning: An Artificial Intelligence Approach. Palo Alto, CA.: Morgan Kaufman. 1983, 1:463-82

[15] Z. Pawlak, "Rough Classification", Journal of Man-Machine Studies 1984, 20: 469-83.

[16] R. Hashemi, F. Choobineh, W. Slikker, and M. Paule, "A Rough-Fuzzy Classifier for Database Mining", The International Journal of Smart Engineering System Design, No. 4, 2002, pp.107-114.

[17] R. Hashemi, A. Tyler, A. Bahrami, "Use of Rough Sets as a Data Mining Tool for Experimental Bio-Data", In Computational Intelligence in Biomedicine and Bioinformatics: Current Trends and Applications, Tomasz G. Smolinski, Mariofanna G. Milanova, and Aboul Ella Hassanien, Editors, Springer-Verlag Publisher, June 2008, pp. 69-91.

[18] L. R. Rabiner, "Readings in Speech Recognition", Chapter A: Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, Morgan Kaufmann Publishers, 1990, pp: 267-296.

[19] X. M. Wang, "Probability Bracket Notation: Markov State Chain Projector, Hidden Markov Models and Dynamic Baysian Networks", CoRR, abs/1212.3817, 2012.

[20] G. D. Forney, "The Viterbi Algorithm", Proceedings of IEEE, 1973, vol. 61, pp.268-278.

[21] W. H. Abdulla, and N. K. Kasabov, "The Concepts of Hidden Markov Model in Speech Recognition", Technical Report TR99/09, Knowledge Engineering Lab, Information Science Department, University of Otago, New Zealand, 1999.