Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 52-13 (2019) 1902–1907

A Markovian Approach for Time Series Prediction for Quality Control

Ahmet Şahin*, Ayşe Dilara Sayımlar*
Zehra Melis Teksan*, Erinç Albey*

**Department of Industrial Engineering, Özyeğin University, İstanbul, 34794, Turkey*
(e-mail: ahmet.sahin@ozyegin.edu.tr).

Abstract: In this work we aim to predict quality levels of incoming batches of a selected product type to a white goods manufacturer from a third party supplier. We apply a Markov Model that captures the quality level of the incoming batch in order to predict the quality status of the future arrivals. The ultimate aim is to generate reliable predictions for the future incoming batches, so that the manufacturing company could warn its supplier if the predictions indicate a significant deterioration in the quality. Applied methodology is compared to several benchmark approaches and its superior performance is shown using a benchmark dataset from the literature and the dataset provided by the manufacturing company. Proposed algorithm performs better compared to benchmarks in detecting the instances with quality level falling outside the tolerances in the validation data; and proves itself as a promising approach for the company.

Keywords: Markov chains, time series, prediction, quality control, industry 4.0

1. INTRODUCTION

In multi-stage manufacturing settings, semi-finished items from external vendors have substantial effect on the end products' overall quality. In most cases, incoming batches contain hundreds of items, that makes impossible to apply quality control tests to every single item. A common technique used in the industry is randomly selecting a set of sample and concluding on the quality of the whole batch. If the average quality level attained from the sample is in certain limits, then the whole batch is sent to the assembly line manufacturing the end product, otherwise, the batch is rejected.

In the setting considered in this work, the dataset is provided by quality control department of Vestel Electronics, one of the leading white goods manufacturers in Turkey. The dataset belongs to rear covers used in television assembly. Since rear covers are not produced in the premises of the company, but outsourced to a third party supplier; the white goods manufacturer does not have process data of the covers.

As far as the past rejections are considered, most of the time, once a batch is rejected then the batches arriving later on tend to be rejected more frequently. This is attributed to the fact that once something starts to go wrong in the supplier's manufacturing process, the fault remains until supplier is warned. As the quality problem persists, consecutive arrivals of faulty batches create starvation in the designated assembly station; hence, severe capacity losses occur in the assembly line. For this reason, the manufacturing company needs to predict timing of faulty batch arrival and inform the supplier to trigger a preventive action on their side.

Once the dataset is considered (details are provided in Section 3), it is seen that the data is single dimensional quality data, and can be seen as a simple time series. The time series are encountered in many production processes in real life. When modelled accurately, substantial impact is realized on prediction and decision processes (Ching et al., 2008). However, time series forecasting has been always a challenging task for researchers and forecast experts due to its very nature (Ky et al., 2018). A lot of methods are suggested to predict time series such as Autoregressive Integrated Moving Average (ARIMA), exponential smoothing, moving averages, and ones that consider seasonality and trends such as Winter's method (Rockwell et al. 2002).

Markov chains, although not frequently comes to one's mind for time series prediction, are in fact effective tools to model time series and are used for time series prediction in the literature. For example, Shamshad et al. (2005) used first and second order Markov chain models to model time series. In our case, the quality values for the selected part calculated as nonnegative real numbers. As far as the work in this paper is concerned, detecting peak values (that is out of limit quality values) is more important than finding the exact quality value. Therefore, continuous quality measurements are converted to categorical values. There are many examples of modelling categorical data sequences using higher-order Markov chain. Raftery (1985) is the first one to suggest using this method. Ching et al. (2004) also applies an enhanced version of Raftery's model to solve similar problems.

In this paper, we apply Markov chain models to predict the quality failures of a part, outsourced to a third party and is used in the final assembly process of TV sets. The paper is organized as follows. Details of the proposed of methodology as well as benchmark approaches can be found in Section 2, which is followed by numerical result discussions in Section 3. Last section, Section 4, presents the concluding remarks.

2. METHODOLOGY

10.1016/j.ifacol.2019.11.480

Although several different techniques can be used to predict time series, in this study we focus on discrete time Markov chain models. We present three different approaches, namely a first order Markov chain model, higher order Markov chain model and proposed algorithm Markov chain model with composite states. Details of these models are presented in the following sub-sections. As a classical benchmark for time series prediction, we also consider ARIMA approach. However, the details of ARIMA is not discussed in this paper. Interested readers could refer to Rockwell et al. (2002).

2.1 First-Order Markov Chain Model

We first consider modelling categorical quality data (or *time series*) by using a first-order Markov chains with k states $E = \{1,2 \dots, k\}$, where each state represents a continuous interval in the quality measurement domain (details are presented in Section 3).

Let's assume that state in period t is represented with $\boldsymbol{X}_t$. In most situations, $\boldsymbol{X}_t$ is not known with certainty before time t and may be viewed as a random variable; the sequence of random variables $\boldsymbol{X}_0, \boldsymbol{X}_1, \boldsymbol{X}_2 \dots$ is defined as a discrete-time stochastic process. A discrete-time stochastic process is called a first order Markov chain, if the following is satisfied:

$$P(\boldsymbol{X}_{t+1} = s_{t+1} | \boldsymbol{X}_t = s_t, \boldsymbol{X}_{t-1} = s_{t-1}, \dots, \boldsymbol{X}_1 = s_1, x_0 = s_0)$$
$$= P(x_{t+1} = s_{t+1} | x_t = s_t) \qquad \forall t, s, \tag{1}$$

where $s_t \in E \ \forall t$ represents the state of the time series at time t. The conditional probabilities,

$$p_{ij} = P(\boldsymbol{X}_{t+1} = s_{t+1} | \boldsymbol{X}_t = s_t) \ \forall i, j \in E, \tag{2}$$

are called the one-step transition probabilities of the Markov chain. The transition probability matrix $\boldsymbol{P}$ can be defined as $\boldsymbol{P} = [p_{ij}]_{k\times k}$. Given that the state at time t is i, the process must be in any state j at time $t+1$. This means that,

$$0 \le p_{ij} \le 1 \ \forall i, j \in E \text{ and} \tag{3}$$
$$\textstyle\sum_{i \in E} p_{ij} = 1, \forall j \in E.$$

A first-order Markov chain model,

$$\boldsymbol{X}_{t+1} = P\boldsymbol{X}_t \tag{4}$$

is then constructed for the observed categorical quality data, where the state probability distribution vector is given by

$$\boldsymbol{X}_{t+m} = [0, \dots, 1(j^{th}\ entry), \dots, 0]^T, \tag{5}$$

if the system is in state $j \in E$ at time $(t + m)$. The proof can be found in (Horn & Johnson, 1985, pp. 508–511).

2.2 Higher-Order Markov Chain Model

In higher-order (say n^{th} order) Markov chain, the state probability distribution at time $t = m + 1$ depends on the state probability distribution of the sequence at times $t = m, m - 1, \dots, m - n + 1$.

The model is given as follows:

$$\boldsymbol{X}_{t+n+1} = \textstyle\sum_i \lambda_i P_i \boldsymbol{X}_{t+n+1-i}, \tag{6}$$

where

$$P_i = P(\boldsymbol{X}_{t+n+1} = s_{t+n+1} | \boldsymbol{X}_{t+i} = s_{t+i}) \ \forall i \in \{1, \dots, n\}, \tag{7}$$

and the weights λ_i should satisfy (8) and (9):

$$\textstyle\sum_i \lambda_i = 1, \tag{8}$$

$$0 \le \lambda_i \le 1 \ \forall i. \tag{9}$$

Considering Equations (6)-(9), following linear programming (LP) model is proposed by Ching et al. (2004) for estimating $\boldsymbol{\lambda}$:

$$\underset{\lambda}{Minimize} \textstyle\sum_{k=1}^m w_k \tag{10}$$

subject to:

$$\begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{pmatrix} \ge \widehat{\boldsymbol{X}} - M \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \end{pmatrix} \tag{11}$$

$$\begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{pmatrix} \ge -\widehat{\boldsymbol{X}} + M \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \end{pmatrix} \tag{12}$$

$$\textstyle\sum_{i=1}^n \lambda_i = 1 \tag{13}$$
$$w_k \ge 0 \ \ \forall k, \lambda_i \ge 0 \ \ \forall i. \tag{14}$$

The LP model presented in equations (8) through (14) aims to minimize the total lags, where w_k denotes lags for data points $k = \{1,2 \dots, m\}$ and λ_i is weights for each higher order transition matrices. $\widehat{\boldsymbol{X}}$ in constraints (11) and (12) represents distribution of data, whereas M is $[P_1\widehat{\boldsymbol{X}} | P_2\widehat{\boldsymbol{X}} | \dots | P_n\widehat{\boldsymbol{X}}]$. Constraint (11) and (12) guarantee that stationary vector of higher order Markov chain is closest to the data distribution. Constraint (13) makes sure that the obtained transition matrix from model is the linear combination of higher order transition matrices, and constraint (14) guarantees nonnegativity of decision variables.

2.3 Markov Chain with Composite States

Another model we used to estimate faulty products is proposed by Carpinone et al. (2015). They suggest a second order Markov chain model that is modelled as a first-order Markov chain. To achieve that, they define the states as composite states of two consecutive periods. Thus, the state space becomes

$$E = \{11, 12, \dots, 1N, 21, 22, \dots, N1, N2, \dots, NN\}. \tag{15}$$

In second-order Markov chain, the transition probabilities are calculated as:

$$P(\boldsymbol{X}_{t+1} = s_{t+1} | \boldsymbol{X}_t = s_t, \boldsymbol{X}_{t-1} = s_{t-1}) \ \text{for} \ \forall s \in E \tag{16}$$

In the new approach, the state of the current period t is denoted by X_{li} and the next period $(t+1)$ by X_{kj}. When we define a one-step transition matrix, the properties of Markov chain forces $i = k$, and the transition probability between two states which does not satisfy that rule becomes 0. The transition probabilities for this case is presented in Equation (17):

$$p_{li,kj} = \begin{cases} P(\boldsymbol{X}_{t+1} = s_{t+1} | \boldsymbol{X}_t = s_t, \boldsymbol{X}_{t-1} = s_{t-1}) \ , & i = k \\ 0, \ otherwise. \end{cases} \tag{17}$$

For example, if we consider a two state Markov chain, the one-step transition matrix can be defined as shown in Table 1.

Table 1. One-step transition matrix.

		kj			
		11	12	21	22
li	11	$p_{11,11}$	$p_{11,12}$	0	0
	12	0	0	$p_{12,21}$	$p_{12,22}$
	21	$p_{21,11}$	$p_{21,12}$	0	0
	22	0	0	$p_{22,21}$	$p_{22,22}$

The probabilities of an n-step matrix can be obtained by taking the n^{th} power of one-step transition matrix. In our example, we calculated a 2-step transition matrix and the related probabilities are represented by:

$$p^{(2)}_{li,kj} = P(\boldsymbol{X}_{t+2} = s_{t+2}, X_{t+1} = s_{t+1} | \boldsymbol{X}_t = s_t, \boldsymbol{X}_{t-1} = s_{t-1}) \quad (18)$$

Table 2 presents how two-step transition matrix can be obtained from a representative Markov chain with two states.

Table 2. Two-step transition matrix.

		kj			
		11	12	21	22
li	11	$p^2_{11,11}$	$p_{11,11} \cdot p_{11,12}$	$p_{11,12} \cdot p_{12,21}$	$p_{11,12} \cdot p_{12,22}$
	12	$p_{12,21} \cdot p_{21,11}$	$p_{12,21} \cdot p_{21,12}$	$p_{12,22} \cdot p_{22,21}$	$p_{12,22} \cdot p_{22,22}$
	21	$p_{21,11} \cdot p_{11,11}$	$p_{21,11} \cdot p_{11,12}$	$p_{21,12} \cdot p_{12,21}$	$p_{21,12} \cdot p_{12,22}$
	22	$p_{22,21} \cdot p_{21,11}$	$p_{22,21} \cdot p_{21,12}$	$p_{22,22} \cdot p_{22,21}$	$p^2_{22,22}$

3. NUMERICAL EXPERIMENT

We consider two different datasets: one from a domain different than manufacturing, namely genetics, and the other from manufacturing domain. The first dataset consists of a DNA sequence (a categorical data sequence of four possible categories showing the introns from the mouse αA-crystallin gene) provided by Raftery et al. (1994). Ching et al. (2004) implemented their model on this dataset (see Figure 1) with different states as suggested by Raftery et al. (1994).

The second set consists of 189-days lab results obtained from quality control tests applied on a selected sample from each arriving batch at each day. The company applies a test for measuring the flatness of the products by selecting parts from a given batch and average of these measurements is checked to see whether the average flatness level is within the required specification limits, that is (0, 0.8). The average values for 189 days are provided in Figure 2.

As mentioned before, raw data needs to be re-organized in order to be used in model fitting and performance testing. The algorithms we use work over a discrete state based data. Since the raw data is continuous, we have to convert raw values into discrete state information. For this purpose, we run a frequency analysis using different intervals and chose the categorical representations that segments the data homogenously over the intervals. After preliminary analysis, we decide to divide data into 3, 5 and 6 states.

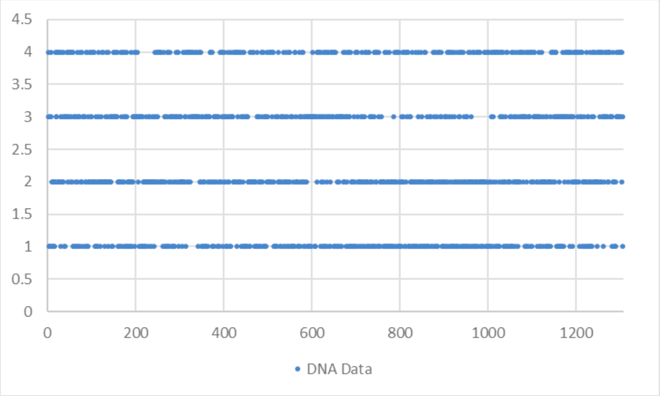

Figure 1. DNA dataset.

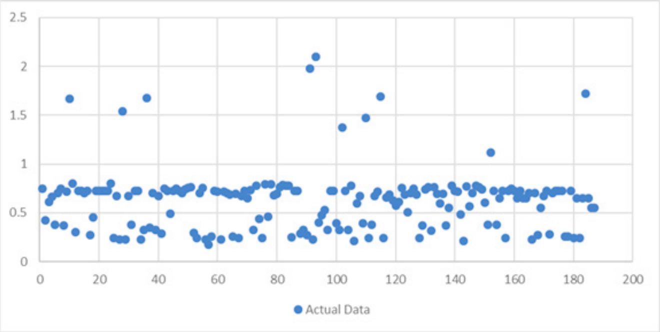

Figure 2. Rear cover dataset.

In the first experiment, the flatness of rear cover measurement can be classified into five possible states (1, 2, 3, 4, 5). The quality series are expressed as (≤0.3), (0.3~0.6), (0.6~0.7), (0.7~0.73), (0.73~0.76) and (≥0.76). For the second, the five states are (≤0.6), (0.6~0.73), (0.73~0.75), (0.75~0.80), and (≥0.80). Lastly, three states are (≤0.5), (0.5~0.75), and (≥0.76).

We split the dataset into training and testing subsets in order to separate the learning and performance testing phases. We learn the transition probabilities for the Markov chains based on first 35 data points (35 is the minimum amount of data required to fit ARIMA model. We keep the initial training dataset the same for all models, to be able to compare the models on a fair ground) and used them to predict the 36th point. Then, we add the actual (real) value of 36th point to our training set, train the models with the new training set again and used it for the prediction of the next point. We repeated this process for all the data points.

We compare the performance of first-order and second-order model with two-step (composite states) first-order and second-order. The comparison is made with different states.

The performance of prediction with Markov chain is measured by prediction accuracy, r, as shown in Equation (19) and (20):

$$r = \frac{1}{N-n} \times \sum_{t=n+1}^{N} a_t \quad (19)$$

where

$$a_t = \begin{cases}1, & x_t = \hat{x}_t \\ 0, & o/w\end{cases}. \qquad (20)$$

In other words, if $\hat{x}_t$, prediction for period t, is equal to x_t, the actual value at period t, then a_t, accuracy for period t is recorded as 1; otherwise as 0. As seen in Equation (19), averaging a_t values over all periods yield the prediction accuracy, namely r value, of the approach.

First, we compare models on DNA dataset used by both Raftery and Ching. We have better results in two-step models than one-step. For 3-State and 4-State models, we get same results because of $\lambda_1 = 1$ and the next step should predict from previous step ($\boldsymbol{X}_{t+3} = P_1\boldsymbol{X}_{t+2}$). This means higher order transition matrix is same with first order transition matrix. Prediction accuracy on DNA dataset is shown in Table 3.

Table 3 Prediction accuracy in the DNA sequence

	2-State model	3-State model	4-State model
Two Step-Second Order	0.59	0.68	0.34
Two Step-First Order	1.00	0.67	0.34
One Step-First Order	0.54	0.49	0.33
One Step-Second Order	0.57	0.49	0.33

As seen in Table 3, for a two state model "one step-first order" approach yields the best accuracy. This is not a surprise due to the fact that next information piece in the DNA sequence data is in fact highly correlated with the previous information. However, prediction performance deteriorates as we increase the number of states. This could be attributed to the fact that as we move on higher degree of states, data needs to be aggregated and the special relationship for the DNA sequence is no more represented in the state space. Once the state number is chosen as four, all four models converge to a similar poor accuracy of 33%.

Secondly, we compare models on rear cover dataset. The main struggle here is to determine the states while converting the data from continuous to categorical data. If different ranges are used, it may also be possible to improve the results. We also forecast the continuous data with ARIMA, then we assign categories to predicted result. The (p=5, d=1, q=2) order of the model for the number of AR parameters, differences, and MA parameters used in ARIMA model. Prediction accuracy on rear cover data is shown in Table 4.

Table 4. Prediction accuracy in the rear cover sequence.

	3-State model	5-State model	6-State model
One Step-First Order	0.51	0.40	0.30
One Step-Second Order	0.51	0.38	0.29
Two Step-First Order	0.47	0.22	0.26
Two Step-Second Order	0.49	0.23	0.27
ARIMA	0.33	0.22	0.13

The actual data points and the points acquired through 6-state models are shown in Figure 3 and for better understanding some values are picked and presented in Figure 4 in higher resolution. As can be seen from the Figures 3 and 4, while one-step models' accuracy rates are high, they cannot predict any of the peak values that represents the defected batches (the values in 6th state). On the other hand, both first-order and second-order models with composite states, predict 10 of the 27 peak values correctly, despite having lower accuracy than one-step models. Some of these points are shown in Figure 4. However, notice that the gap between accuracy rates is not very high. ARIMA, on the other hand, reaches an accuracy rate lower than the accuracy rate could be achieved if the next period's state is chosen randomly. The model predicts 16 peak values. However, seven of these values are predicted while the actual state is 5, and, the rest while the actual state is lower. The ARIMA model realizes it should catch some peak values but cannot increase the predicted value enough and end up in one state lower than the actual peak value. The 84th sample point is a good example for its representation.

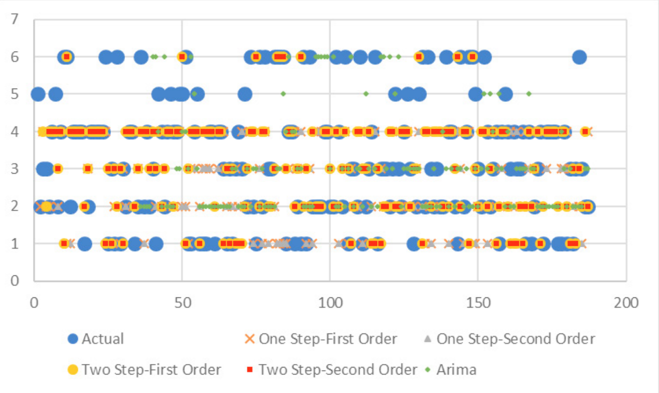

Figure 3. Comparison of predictions and actual for State 6.

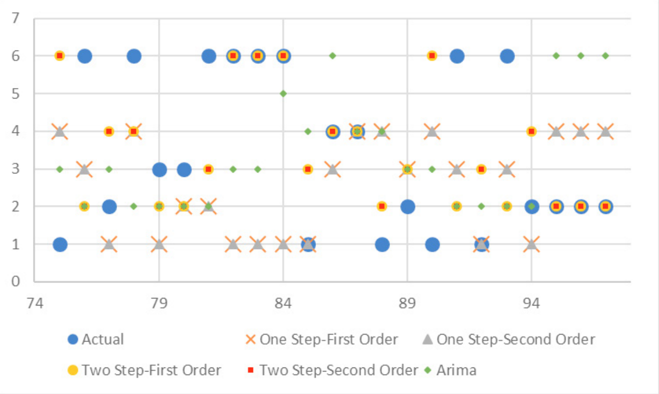

Figure 4. Comparison of predictions and actual for State 6 (detailed).

When 5-state models are considered the accuracy rate of one-step models are much higher than two-step models. Nevertheless, the one-step models, again, are not successful at predicting any peak points, which are the points in 4th and 5th state, while both two-step models predicted six out of 27 peak points correctly, as can be seen in Figure 5. None of the models is successful at predicting peak values at state 5. The prediction performance for 4th state peak points can be seen in Figure 6, which demonstrates important points of Figure 5 in higher resolution. ARIMA achieves a relatively higher performance than its 6-state model in terms of accuracy rate. It predicts 21 peak values, however, all of them are incorrect. 11 of these predictions' actual state is 3, and the rest is lower.

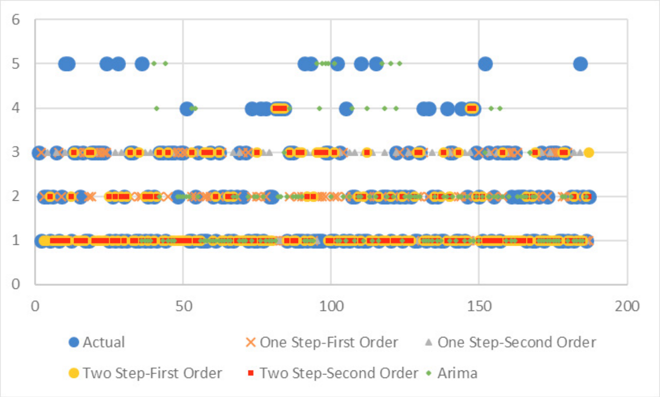

Figure 5. Comparison of predictions and actual for State 5.

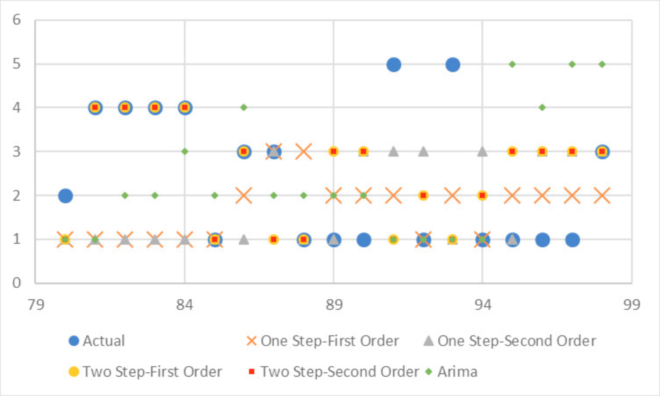

Figure 6. Comparison of predictions and actual for State 5 (detailed).

In the 3-state models, the accuracies of all Markov chain models are very close to 50% while one-step models have slightly higher values. ARIMA, on the other hand, has an accuracy rate that could be achieved with random selection. Again, the one-step models fail to predict any values in the 3rd state, which represents faulty batches. The first-order two-step model predicted 2 of the 36 peak values, and the two-step first-order model predicted 3 of these values correctly (See Figure 7 and for a finer representation see Figure 8). ARIMA predicts 16 peak points but all of them are predicted while the actual state is 2.

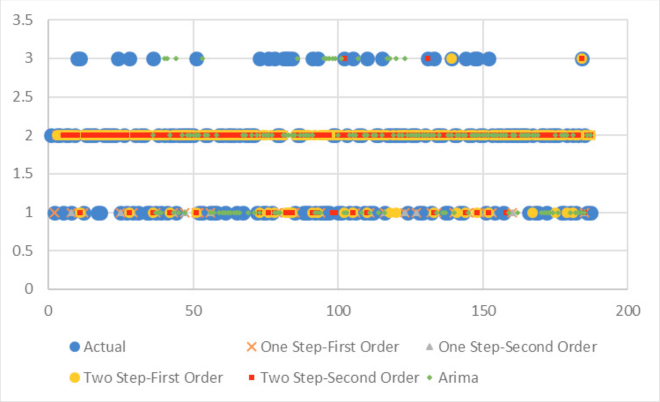

Figure 7. Comparison of predictions and actual for State 3.

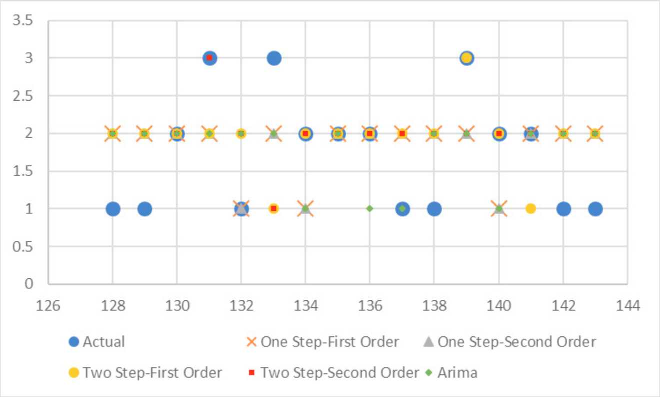

Figure 8. Comparison of predictions and actual for State 3 (detailed).

4. CONCLUSIONS

In this paper, we applied several different discrete time Markov models, with varying state definitions, to analyze and predict the quality values of a sub-assembly part used in TV manufacturing. Proposed models are applied to an existing dataset concerning DNA dataset and rear cover dataset and numerical results are given. The results show that the performance of Markov chain models to predict the quality data is superior to that of ARIMA model.

As a matter of course, the approach proposed in this paper requires further validation with other datasets. Once the validation is approved, the company envisions a decision support system, which allows to monitor all incoming sub-products from third party suppliers and provided the decision makers with predictions of all incoming batches for a pre-determined period (say a week) of time. This way, it will be possible for the company to control its supply chain closer and will be easier to take proactive actions such as warning the suppliers before faulty batches are produced. A natural future direction will be enriching the dataset with the process data, such as machine conditions at the time of production (temperature, vibration values of critical components, electricity consumption, current values etc.) and environmental conditions, such as humidity and temperature, of the suppliers' manufacturing facility. Once dataset is enriched, more complex learning approaches could be proposed to improve prediction accuracy.

REFERENCES

Carpinone, A., Giorgio, M., Langella, R., & Testa, A. (2015). Markov chain modeling for very-short-term wind power forecasting. *Electric Power Systems Research*, *122*, 152-158.

Ching, W. K., Fung, E. S., & Ng, M. K. (2004). Higher-order Markov chain models for categorical data sequences. *Naval Research Logistics (NRL)*, *51*(4), 557-574.

Ching, W. K., Ng, M. K., & Fung, E. S. (2008). Higher-order multivariate Markov chains and their applications. *Linear Algebra and its Applications*, *428*(2-3), 492-507.

Ky, D. X., & Tuyen, L. T. (2018). A Higher order Markov model for time series forecasting. *International Journal of Applied Mathematics and Statistics™*, *57*(3), 1-18.

Raftery, A. E. (1985). A model for high-order Markov chains. *Journal of the Royal Statistical Society. Series B (Methodological)*, 528-539.

Raftery, A., & Tavaré, S. (1994). Estimation and modelling repeated patterns in high order Markov chains with the mixture transition distribution model. *Applied Statistics*, 179-199.

Shamshad, A., Bawadi, M. A., Hussin, W. W., Majid, T. A., & Sanusi, S. A. M. (2005). First and second order Markov chain models for synthetic generation of wind speed time series. *Energy*, *30*(5), 693-708.

Brockwell, P. J., Davis, R. A., & Calder, M. V. (2002). *Introduction to time series and forecasting* (Vol. 2). New York: Springer.