A novel autonomous machine learning technique for recognizing control chart patterns

Raghavendra Ramesh* | Surendra Yadav† | Ashendra Kumar Saxena‡

*Jain (deemed to be) University, Bangalore, India, Assistant Professor, Department of Computer Science and Information Technology.
†Vivekananda Global University, Jaipur, India, Professor, Department of Computer Science & Application.
‡Teerthanker Mahaveer University, Moradabad, Uttar Pradesh, India, Professor, College Of Computing Science And Information Technology.

Abstract A number of strategies have been developed for controlling and monitoring the production process since the quality of the product has emerged as one of the key concerns in today’s manufacturing sector. Control charts are the best tools for monitoring and adjusting products and processes. This research proposes a novel automatic method for the recognition of nine control chart patterns (CCPs) based on novel autonomous machine learning. The classification portion and the tuning portion make up the two main components of this procedure. Support Vector Machine (SVM) have demonstrated outstanding performance over the past few years on a variety of applications, including signal dispensation, speech detection, and image processing. SVM is consequently employed as the intelligent classifier for the recognition of CCPs in the classification phase. One key challenge with SVM is that it requires a high level of expertise to choose appropriate parameters, such as the quantity of kernel and their spatial diameters, knowledge rate, etc. It is difficult to fine-tune the SVM parameters because of their domestic dependence. These problems led to the employment of the Harmony Search (HS) Algorithm for the best tuning of SVM parameters in the tuning section of the proposed technique. Instead of depending on any feature engineering procedures, the suggested method, in contrast to the popular CCPs recognition methods, takes raw data and runs it through many hidden layers to obtain the best feature representation. The quantitative and simulation results demonstrate the suggested method’s performance advantage over the earlier methods.

Keywords: support vector machine, harmony search algorithm, control chart patterns, novel autonomous machine learning, signal processing

1. Introduction

The Autonomous Machine Learning (AutoML) method is to eliminate the need for considerable human involvement in the creation and rollout of ML models. It is difficult to single out a single "novel" AutoML approach since the subject is developing at such a fast pace (Xie et al 2019) Since neural networks form the backbone of many machine learning models, their construction may be automated through a method called Neural Architecture Search (NAS). NAS seeks to automate the process of designing neural network architectures, which is currently done either via trial and error or with specialized expertise (Zhang et al 2019). Various layers, connections, hyper parameters, and activation functions are all part of the architectural space that is being searched. The method considers much potential architecture, each of which is trained and then tested on a validation dataset. This is often a very resource-intensive and time-consuming activity on a computer (Miikkulainen et al 2019). Agent trained using reinforcement learning may pick and construct novel designs by maximizing a reward signal that represents the architecture's performance on a specific task. To develop better designs, the agent repeatedly samples them, trains and assesses them, and changes its policy (She et al 2020) Using evolutionary algorithms to probe the architectural space. Keeping an architectural population alive, evolving it via processes like mutation and crossover, and measuring its effectiveness are all part of this process. In an evolutionary process, the best structures tend to procreate and spread across the population (Khan et al 2019). To solve the architectural search problem, the gradient-based NAS method poses it as a continuous optimization issue. To enable gradient-based optimization, it employs continuous relaxation methods in place of discretely selecting architecture. Through repeated iterations, the search method optimizes a differentiable surrogate goal by modifying architectural characteristics like channel widths and layer sizes (Sun et al 2021). To effectively explore the design space, a Bayesian optimization-based NAS technique uses this method. It employs Bayesian approaches to efficiently search for and identify potential designs by modelling their performance as a surrogate function (Loquercio et al 2020) The purpose of NAS is to eliminate the need for human design and experimentation by automating the architectural search process and discovering neural network designs that deliver high performance on certain tasks (Dang et al 2020). It's crucial to remember that the area of AutoML is always changing, and new methods are being created to handle
certain problems and enhance the speed and efficacy of the creation of machine learning models (Garcia et al. 2020). An important part of statistical process control's (SPC's) monitoring and analysis of process variability is identifying trends in control charts. It is possible to automate the detection and categorization of control chart patterns using machine learning methods. One of these methods is supervised learning, in which an algorithm learns to detect various control chart patterns by seeing labelled instances (Sejnowski 2020). Priming the data you need a labelled dataset that includes control charts with identified patterns to use supervised learning. This data collection has to have a number of control charts, all of which are clearly labelled with their respective patterns (normal, shift, trend, cyclic, etc.). The control charts may be created using either actual or simulated process data (Jahangir et al. 2020).

2. Related Works

Research (Rajula et al. 2020) evaluate ML and classical statistical approaches for their strengths and weaknesses in the context of healthcare. When a priori knowledge of the issue at hand is significant, as it often is in public health research, and the number of cases much surpasses the number of variables under examination, traditional statistical approaches seem to be more beneficial. Beginning with a conceptual model of the pipeline of ML-HCAs from inception to development to deployment and the parallel pipeline of review and supervision duties at each step, this study presents a systematic strategy to detecting ML-HCA ethical problems. We build upon this framework by asking central questions that bring up ethical concerns and by recognizing ethical problems that have received little to no prior attention but which are still important (Char et al. 2020). The study (Usama et al. 2019), we will take a high-level look at how unsupervised learning has been put to use in the field of networking. We provide a thorough overview of the most recent developments in unsupervised learning methods and illustrate their applicability in a variety of networking-related learning problems. The study provides a brief overview of the development and categorization of machine learning, before delving into the cutting-edge of this field as it has been applied to Unmanned Aerial Vehicles (UAV) for autonomous flying. Several methods of control, such as tuning parameters, adaptive control in an uncertain environment, route planning in real time, and object identification, are provided (Choi and Cha 2019). Applying machine learning methods, as shown in this research, has improved smart farming's precision. Fruits and vegetables such grapes, apples, oranges, and tomatoes, as well as cereal grains like corn and wheat, were harvested for the research. The paper's study results are commercially accessible as tools for a variety of uses, including automated harvesting, weed identification, and insect control (Darwin et al. 2021). In the study (Ferdowsi et al. 2019) offer edge analytics architecture for ITSs, where data processing occurs at the level of the vehicle or roadside smart sensor, to address the latency and reliability issues plaguing ITSs. For more accurate mobile sensing in ITSs, a distributed edge computing architecture may take use of deep learning methods by combining the power of passengers’ mobile devices with in-vehicle processors (Ferdowsi et al. 2019). In the paper, they provide a long-term deep learning architecture that effectively classifies objects by fusing and selecting deep features from several layers of data. The strategy presented consists of three stages: very deep convolution networks for large-scale image recognition and inception V3 are used to extract features via transfer learning; (2) feature vectors are fused using a parallel maximum covariance approach; and (3) the best features are selected via a multi logistic regression-controlled entropy-variances method (Rashid et al. 2020). In the study (Chen et al. 2019) create a multi-level Deep Reinforcement Learning (DRL) protocol for acquiring knowledge about lane-changing habits in heavy traffic. Faster and safer lane changes may be taught by first dissecting the overall behavior into its component sub-policies. The paper presents a thorough analysis of ML applications across several additive manufacturing (AM) fields. ML may be used to generate cutting-edge met materials with improved performance and refined topological layouts in the design for additive manufacturing (DfAM). Modern ML algorithms may aid in optimizing AM process settings, analyzing powder dispersal, and detecting defects in real time (Wang et al. 2020).

3. Methodology

In this research, a novel SVM-based hybrid approach to recognizing nine CCPs is suggested. In contrast to traditional, hand-crafted feature representation approaches, the suggested approach learns feature representation automatically from the raw training data. The suggested method involves feeding raw data into SVM and having it identify the CCPs present. To successfully apply SVM to a new project, one must have extensive understanding of the method and be able to choose the appropriate parameters, including activation function, knowledge rate, kernel sizes, and the number of kernels. These parameters have deep reliance on other values, making fine-tuning them a costly endeavour. A larger searching area may damage training, while increasing the number of small-sized kernels improves performance. Reduced processing costs and less overlapping receptor fields are two additional advantages of using the optimum stride length. This means picking the right value for these variables.

3.1. Support vector machine

https://www.malque.pub/ojs/index.php/msj
The support vector machine (SVM) is a supervised ML approach that is a relatively new and promising classification model based on the idea of structural risk reduction. The margin is maximized and strong generalization capacity is attained by using a separating hyperplane in this model.

\[ f(x) = (w, \phi(x)) + b \]  
(1)

Where \( \phi(x) \) is a function from the input space to a high-dimensional feature space; \( w \) is a coefficient vector; and \( b \) is the offset of the hyperplane from the origin; these values are found by solving the optimization function that follows.

\[ g(w, \xi) = \|w\|^2 + c \sum_{i=1}^{N} \xi_i \]  
(2)

\[ y_i \left( (w, \phi(x_i)) + b \right) \geq 1 - \xi_i, \xi_i \geq 0 \]  
(3)

The following formula yields a kernel function, where \( \epsilon_i \) is the slack variable, \( c(\epsilon > 0) \) the regularization variable of the errors, and \( (\epsilon > 0) \) the zero-the order constant:

\[ k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle \]  
(4)

The \( x_i \) element of the training sample vectors is the kernel function \( k \). Linear, polynomial, radial basis function (RBF), and sigmoid kernels are some of the many types of kernels used in support vector machines. SVM has the potential to outperform other kernel functions when it comes to non-linear classification. This research is to look into SVM in the context of bagging, boosting, and stacking models as component classifiers.

3.2. Harmony search (HS)

We present a metaheuristic method called harmony search (HS) that uses a population-based strategy. The concept of harmony seek was developed in part as an homage to the art of improvising music. Each element of the solution vector is compared to a note in a musical scale, and the algorithm mimics the way a musician would go about adjusting the notes to get a desired effect. However, the convergence speed of basic HS is lower, and it may have issues with local minima. Five primary factors are presented in Table 1 and five primary processes are outlined below for the fundamental HS. We propose an enhanced HS method in this work by fusing HS approaches with extreme decomposition. Crossover rate (CR) and scaling factor (F) are two new metrics added in DE that stand for these two concepts. Figure 1 provides a high-level overview of the design, and the essential procedures are described in more depth below.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMCN</td>
<td>Rate of recollection harmony</td>
</tr>
<tr>
<td>MaxImp</td>
<td>Maximum number of improvisations</td>
</tr>
<tr>
<td>bw</td>
<td>Bandwidth vector</td>
</tr>
<tr>
<td>PAR</td>
<td>Change in pitch per second</td>
</tr>
<tr>
<td>HMS</td>
<td>Size of the symmetric group's memory, in terms of the number of solution vectors</td>
</tr>
</tbody>
</table>
Procedure for Getting Started All algorithm parameters, including CR, F, HMS, HMCR, PAR, MaxImp, and bw, must be specified. The harmony memory is then set to a random value as:

\[
HM = \begin{bmatrix}
    x^1 \_1 & x^1 \_2 & \ldots & x^1 \_N \\
    x^2 \_1 & x^2 \_2 & \ldots & x^2 \_N \\
    \vdots & \vdots & \ddots & \vdots \\
    x^HMS \_1 & x^HMS \_2 & \ldots & x^HMS \_N \\
\end{bmatrix} = f(X^1) 
\]

(5)

For \( j = 1, 2 \ldots \) HMS, a solution vector \( X_j = (x^j \_1, x^j \_2, \ldots x^j \_N) \) is defined, where \( f(X) \) is the value of the objective function at \( X \) and \( x^j \_i \) is the \( i \)th value of the \( j \)th solution vector. Finding \( X_{\text{best}} \) and \( X_{\text{worst}} \) requires computing the fitness of the objective function.

\[
bw_i(\text{Imp}) = \frac{x^U \_i - x^L \_i + 0.002}{10} \cdot \exp \left( -10 \cdot \frac{\text{Imp}}{\text{MaxImp}} \right) 
\]

(6)

Where Imp is the total number of improvised pieces and \( x^U \_i \) and \( x^L \_i \) are the upper and lower boundaries of \( x \). We develop the algorithm 1 presented in the following paragraph to build a new harmony. For example, \( r \) and\((L, U)\) returns a random number between \( L \) and \( U \) using the normal distribution, whereas \( r \) and\((l, U)\) returns a random number between \( l \) and \( U \) using the uniform distribution.

**Algorithm 1. Improvisational operation in HS pseudo code**

for each \( i \in \{1, 2 \ldots N\} \) do
    if rand\((0, 1)\) \leq HMCR then
        if rand\((0, 1)\) \leq PAR then
            \( x_{\text{new},i} = x^\text{best}_i + \text{bw} \times \text{rand}(-1,1) \);
        else
            \( x_R = 2 \times x^\text{best}_i - x^\text{worst}_i \);
            if \( x_R > x^U_i \) then
                \( x_R = x^U_i \);
                end if
            end if
        end if
    else
        \( x_{\text{new},i} = x^\text{worst}_i + \text{rand}(0,1) \times (x_R - x^\text{worst}_i) \);
    end if
end for

---

**Figure 1** Example of the HS algorithm’s flowchart.
end if
else
    \( x_{\text{new},i} \leftarrow x_{i} + \text{rand}(0,1)^* (x_{i}^\text{U} - x_{i}^\text{L}) \);
end if
end for
ReturnX_new

4. Result and Discussion

In this study, we conducted a number of experiments and compared the outcomes of the suggested approach to those obtained using other methods. MATLAB, installed on a Windows 10 64-bit professional PC with 64 GB of RAM, is used for both the SVM's development and analysis. Five independent runs of the suggested method and alternative classifiers are demonstrated, with the average efficiency demonstrated.

4.1. Evaluation of the proposed technique's performance on a dataset

One thousand examples of each pattern are generated using the provided equations. To conduct a performance study of the suggested approach, each equation represents the relationship between the standard normal variate value \( r_i \) and the 'observed value' \( y_i \) at the \( i \)-th time point. For a 'normal process with mean' \( \mu \) and 'standard deviation' \( \sigma \), the following formulae may be used to generate a wide range of 60-point patterns:

Systematic pattern:

\[
y_i = \mu + r_i \sigma + d \times (-1)^i, 1 \sigma \leq d \leq 3 \sigma \quad (7)
\]

Normal pattern:

\[
y_i = \mu + r_i \sigma', \mu = 80, \sigma = 5 \quad (8)
\]

Combine pattern:

\[
y_i = \mu + r_i \sigma + (-1)^w m, 1.5 \sigma \leq m \leq 2.5 \sigma \text{ and } w \text{ is } 1 \text{ or } 0 \quad (9)
\]

Stratification pattern:

\[
y_i = \mu + r_i \sigma', 0.2 \sigma \leq \sigma' \leq 0.2 \sigma \quad (10)
\]

Where 'w' is a binary integer value that depends on the transition probabilities between distributions, which are described by the parameters \( b = mp \text{ and } p \) \((0 < p < 1)\). Since \( b \) is always set to 0.4, this means that we always have \( w = 0 \) if \( p < 0.4 \) and \( w = 1 \) if \( p \leq 0.4 \).

Cyclic pattern:

\[
y_i = \mu + r_i \sigma + \sin(2\pi/T), 1.5 \sigma \leq a \leq 2.5 \sigma \text{ and } 8 \leq T \leq 16 \quad (11)
\]

The amplitude of cyclic variation, denoted by 'a', and the period of a cycle, denoted by 'T', are both shown in this equation.

Increasing Trend pattern:

\[
y_i = \mu + r_i \sigma + ig, 0.05\sigma \leq g \leq 0.1\sigma \quad (12)
\]

Reduced trend pattern:

\[
y_i = \mu + r_i \sigma + ig, 0.1\sigma \leq g \leq 0.05\sigma \quad (13)
\]

The amplitude of the gradient for the trend patterns is represented by 'g' in Equations (11) and (12).

Upward Shift pattern:

\[
y_i = \mu + r_i \sigma + ks, k = 1i f i \geq P, \quad (14)
\]

Downward Shift pattern:

\[
y_i = \mu + r_i \sigma - ks, k = 1i f i \geq P, \quad (15)
\]

Parameter 'k' determines the location of the shift, while 's' denotes the magnitude of the shift, in Equations (13) and (14). 9000 samples have been created [Input 10000]. To demonstrate the effectiveness of the suggested system, a split is made between the training and testing phases, each using half of the data.

4.2. Evaluation of the success of the intended method

https://www.malque.pub/ojs/index.php/msj
The effectiveness of the suggested method is studied in this section. The SVM architecture may be represented by the parameterized functions. We utilized the HS method to get the best possible settings for all of the parameters. The maximum number of repetitions is set at 100, while the population size (in terms of hawks) is restricted to 25. When utilizing the HS method, a SVM with five hidden layers and a knowledge rate of 0.00126 performs the best. Table 2 displays the best SVM ideal parameter settings.

<table>
<thead>
<tr>
<th>Layer</th>
<th>Layer type</th>
<th>Stride(S)</th>
<th>Zero padding (p)</th>
<th>No. of trainable parameters</th>
<th>Output shape</th>
<th>Kernel Size (F)</th>
<th>No. of Kernel (K)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>CONV</td>
<td>3</td>
<td>3</td>
<td>961</td>
<td>7×21</td>
<td>5×2</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>POOL</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>8×21</td>
<td>3×2</td>
<td>21</td>
</tr>
<tr>
<td>1</td>
<td>CONV</td>
<td>2</td>
<td>2</td>
<td>2561</td>
<td>5×2</td>
<td>5×2</td>
<td>33</td>
</tr>
<tr>
<td></td>
<td>POOL</td>
<td>3</td>
<td>-</td>
<td>-</td>
<td>7×34</td>
<td>3×2</td>
<td>33</td>
</tr>
<tr>
<td>2</td>
<td>CONV</td>
<td>2</td>
<td>3</td>
<td>289</td>
<td>9×2</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>POOL</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>20×9</td>
<td>5×2</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>CONV</td>
<td>2</td>
<td>2</td>
<td>385</td>
<td>5×2</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td></td>
<td>POOL</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>17×13</td>
<td>4×2</td>
<td>13</td>
</tr>
<tr>
<td>4</td>
<td>CONV</td>
<td>2</td>
<td>2</td>
<td>49</td>
<td>13×2</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>POOL</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>47×4</td>
<td>7×2</td>
<td>5</td>
</tr>
</tbody>
</table>

There are sixty neurons in the input layer, and four additional convolution and pooling layers with four, eight, twelve, twenty, and thirty-two filters in the hidden layer. Which demonstrates the excellent classification accuracy of the proposed method (HS-SVM) using ELU activation functions. In this analysis, the efficiency of a Support SVM built on the same principles but using the ReLU activation function is calculated. The Table 3 below shows the outcomes of utilizing SVM with the ELU and ReLU activation functions.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Accuracy (%)</th>
<th>Standard Deviation</th>
<th>Activation function</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM with optimal architecture</td>
<td>98.81</td>
<td>98.81</td>
<td>98.81</td>
</tr>
<tr>
<td>SVM with optimal architecture</td>
<td>98.12</td>
<td>98.76</td>
<td>98.56</td>
</tr>
</tbody>
</table>

4.3. Analysis in contrast to other classifications

Several tests have been conducted to analyse the performance of the proposed method in comparison to existing machine learning methods. Probabilistic neural networks (PNN), multilayer perceptron neural networks (MLPNNs) using a variety of training methods (including “back propagation” (BP), “resilient propagation” (Rprop), and “Levenberg Marquardt” (LM), as well as “random forest” (RF), are all taken into account for this task. Different classifiers are fed the raw data in these tests, with the same training/testing split. Three important MLPNN setup options are the number of hidden layers, the knowledge rate, and the kind of transfer function. The effectiveness of an RBFNN is very sensitive to the amount of radial basis functions used and the distributions of those functions. Choosing a suitable spread value in PNN is a crucial step in the process. Analysing the suggested classifier against other classifiers on raw data and comparing their performance shows in Figure 2. Accurate identification of the radii value, membership function type, and fuzzy inference system is required for successful ANFIS implementation. For optimal performance in SVM, it is essential to choose a suitable kernel function, kernel parameters, and penalty parameters. The density of foliage has a significant effect on RF signal clarity and performance. To determine what those values should be, the HS method is employed in this experiment. Density of foliage has a significant effect on RF signal clarity and performance. To determine what those values should be, the HS method is employed in this experiment.
4.4. Analysing and contrasting the various methods presented in the research

One of the most common and useful tools for keeping an eye on output in many industries is the control chart. As a result, this strategy is widely used in the industrial sector, and a wealth of research has been conducted on the topic. The effectiveness of authors’ methods has been studied across a variety of datasets. Since no single dataset exists under which all possible approaches can be evaluated, direct comparisons between them are impossible. Studies also vary in their sample sizes and the types of patterns they examine. Most studies on the control chart idea have focused on studying preexisting patterns, such as normal, cyclic, trend, and shift patterns. Because of this, classifying patterns into groups of eight or nine is outside the scope of the suggested approach. And because some of the feature extractions need for designer input, their CCP identification method is not really automated. For instance, the user must decide where to draw the line when dividing a pattern into two displays. The suggested approach achieving convergence over a variety of runs displays in Figure 3.

5. Conclusion

Given the cutthroat nature of the marketplace, quality control and monitoring have emerged as major concerns for manufacturers. In this research, an automated approach to CCPs recognition is suggested, one that makes use of SVM and an optimization technique. The suggested technique allows for the automated recognition of the nine most often observed CCPs without the need for any hand-crafted characteristics. Furthermore, the suggested approach may be used without modification to an unlimited number of CCPs. Multiple tests were conducted, and HS-SVM was compared to numerous
different techniques in order to evaluate its quality and performance. In these simulations, HS is used to determine the best configuration for a SVM and its associated parameters. The collected findings demonstrated the high accuracy of the suggested method and its ability to categorize the nine CCPs with a 99.80 percent success rate. The findings show that the suggested technique is better than the state-of-the-art classifiers, including MLPNN, RBFNN, ANFIS, RF, and SVM. In addition, the suggested strategy outperforms previously published findings in the literature in terms of classification accuracy. Based on the findings, this research strongly suggests using HS-SVM for CCPs identification. The suggested approach may also be used to classify heartbeats, identify breast cancer tumour types, etc., all of which are examples of complex pattern recognition issues.

**Ethical considerations**

Not applicable.

**Declaration of interest**

The authors declare no conflicts of interest.

**Funding**

This research did not receive any financial support.

**References**


