Volume: 1

Issue: 2

Year: 2020

ISSN: 2717-8226

https://dergipark.org.tr/en/pub/jscai

Contents

Journal of Soft Computing and Artificial Intelligence
JSCAI, 1(2): pp:59-68 (2020), www.dergipark.org.tr/en/pub/jscai
JSCAI
ISSN: 2717-8226
Received: August 21 2020 Accepted: September 6 2020

Estimation of Cutting Forces Obtained by Machining AISI 1050 Steel with Cryo-Treated and Untreated Cutting Insert by Using Artificial Neural Network

[1]Şehmus Baday , [2]Onur Ersöz

[1]Department of Mechanical and Manufacturing Engineering, Batman University, Turkey
[2]Institute of Science, Batman University, Turkey

Corresponding author: Şehmus Baday (sehmus.baday@ batman.edu.tr).

ABSTRACT Cutting force is one of the most important criteria for evaluating machinability of workpieces. For this purpose, in present study, prediction of the cutting forces obtained by turning AISI 1050 steel with cryo-treated and untreated CVD-coated cutting inserts with artificial neural networks (ANN) was investigated. Machining parameters such as feed rate, cutting speed and conditions of cutting insert were selected. These parameters were used as input parameters while cutting force was used as output parameter. The employed ANN structure was chosen according to network type, training function, adaption learning function and performance function as feed-forward back propagation, TRAINLM, LEARNGD and MSE, respectively. Thus, the estimation values of the cutting forces attained from ANN model during training and experimental values coincide perfectly with the regression lines, which make the R^2 = 0.99874 in training. For this reason, cutting force was explained by ANN with an acceptable accuracy in this study.

KEYWORDS: ANN, Cutting force, Cutting insert, Cryogenic heat treatment

1. INTRODUCTION

The ANN has been applied to many areas such as prediction of cutting forces, surface roughness, vibration, tool wear, and tool temperature etc. The cutting force is one of the most important criteria for interpreting machinability of workpieces. Cutting parameters such as depth of cut, cutting speed, feed rate and tool toughness are the most effective parameters on the cutting forces [1-3], which are changed with heat treatment processes that applied cutting insert. Cryogenic heat treatment is one of them for improving cutting tool toughness. Therefore, deep cryogenic heat treatment is applied the CVD-coated cutting inserts. Some studies on cutting forces are given below:

Baday [4] investigated prediction of cutting force, obtained during turning AISI 1050 steel, which is applied heat treatment, by using ANN. In the ANN model, feed-forward back propagation as the network type, TRAINLM, BFGS and SCG as the training function, LEARNGD as the adaption learning function, one hidden layer with 10 to 15 neurons, was selected to obtain the best R2 result. Hanief et al. [5] studied modeling and prediction of cutting force, obtained from turning of red brass, by using ANN and regression analysis. They stated that the regression model is able to estimate the cutting force with high accuracy. Nevertheless, the estimation of ANN structure is much better than that of regression model. Gürbüz et al. [6] investigated the effect of cutting inserts with different chip breaker forms on cutting force and modeled them with mathematical formulizations. They analyzed cutting force with 10 different mathematical models for different chip breaker forms. They mentioned that of the mathematical models, polynomial modeling was obtained with the best result, while with Fourier model, they obtained the worst results. Başak et al. [7] analyzed cutting force and surface roughness with

regression and ANOVA analyses. They observed that feed rate was the most significant parameter on surface roughness according to ANOVA analyses. They attained R^2 = 94.6, 94.2% values for cutting force and surface roughness, respectively. Yağmur et al. [8] investigated cutting force during milling of carbon fiber reinforced composite materials via mathematical modeling and they evaluated it. They revealed that increasing cutting speed positively affected cutting forces. Furthermore, the effective parameters on cutting forces were analyzed according to the ANOVA. Ulas et al. [9] studied prediction of cutting forces obtained during turning AISI 304 (Austenitic), AISI 420 (Martensitic) and AISI 2205 (Duplex) stainless steels using ANN techniques. They found that prediction of cutting force with ANN and experimental data were very close to each other. Uzun et al. [10] examined the effect of mechanical properties of AISI 5140 steel on cutting forces, obtained during turning workpiece. They observed that when cutting speed increased, generally, cutting forces obviously decreased. Özkan et al. [11] examined the prediction of cutting forces attained from different cutting conditions during turning operation by using ANN model. They determined that the implemented ANN model was in good agreement with estimation of cutting forces. Kurt et al., [12] developed a mathematical model for estimation of cutting forces. Depending on cutting parameters such as depth of cut, feed rate, cutting speed, and rake angle of the chip breaker of cutting insert, they performed estimation of cutting forces by using regression analyses, which is a statistical analysis method. Jeyakumar et al. [13] investigated the impact of machining parameters such as feed rate, depth of cut, spindle speed and nose radius on the cutting force. They predicted cutting forces with response surface method considering machining parameters. They found that predicted and experimental cutting force values were in good agreement with each other. Kara et al. [14] utilized ANN in their study to predict the main cutting forces, which are orthogonally obtained from machining AISI 304 stainless steel. They received input parameters in ANN model such as feed rate, cutting speed and coating type. They found that the prediction of cutting force value with ANN and experimental values were in good agreement with each other. Asiltürk et al. [15] carried out the estimation of cutting forces by using ANN model in turning 4140 steel considering feed rate, cutting speed and depth of cut. They indicated that the implemented ANN model had good performance for prediction of cutting force.

In the view of the literature, many researchers investigated prediction of cutting forces considering cutting parameters using ANN, mathematical model and experimental methods. However, prediction of cutting forces obtained from cryogenically heat-treated CVD-coating cutting insert has not been come across in the literature. For this purpose, in this study, prediction of cutting forces attained from machining AISI 1050 steel with CVD-coating insert by using ANN was investigated.

2. MATERIALS and METHOD

In this study, workpiece material was chosen as AISI 1050 steel bar with a diameter of 60 mm, which was turned 1.5 mm depth of cut for longitudinal turning in this experimental. Cutting insert was selected as CVD coated WNMG with MP, which is medium chip breaker, for using generally medium carbon steel in machining. This cutting insert is given in Figure 1. Machining experiments were carried out on CNC turning machine in accordance with cutting parameters, which were specified depending on the knowledge of manufacturing procedures, which are advised as ideal conditions in medium carbon steel. Therefore, cutting speed was selected as 200, 220 and 240 m/min, and feed rates was selected as 0.1, 0.2 and 0.3 mm/rev. The depth of cut was remained constant as 2 mm in the machining experiments. A new cutting insert was used in each machining experiment to obtain the same cutting condition. Main cutting force was measured using KISTLER Type 5070 dynamometer, which obtains three-component piezoelectric.

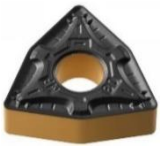


Figure 1. CVD-coated WNMG-MP carbide turning insert

2.1. Deep Cryogenic Heat Treatment

According to literature review, the holding time of cryogenic heat is attained 24 hours in terms of cutting tool wear [16, 17]. In this study, deep cryogenic heat treatment was subjected to CVD coated cutting insert in three stages: In first stage, cutting inserts were cooled down slowly at the temperature, which came down from room temperature to deep cryogenic treatment (-146 ° C) in 2 hours. In the second stage, inserts were hold at -146 ° C for soaking time namely for 24 hours and then brought back to room temperatures. Finally, they were tempered at 200 ° C two times oscillating, which lasted for 2 hours. All stages are shown in Figure 2.

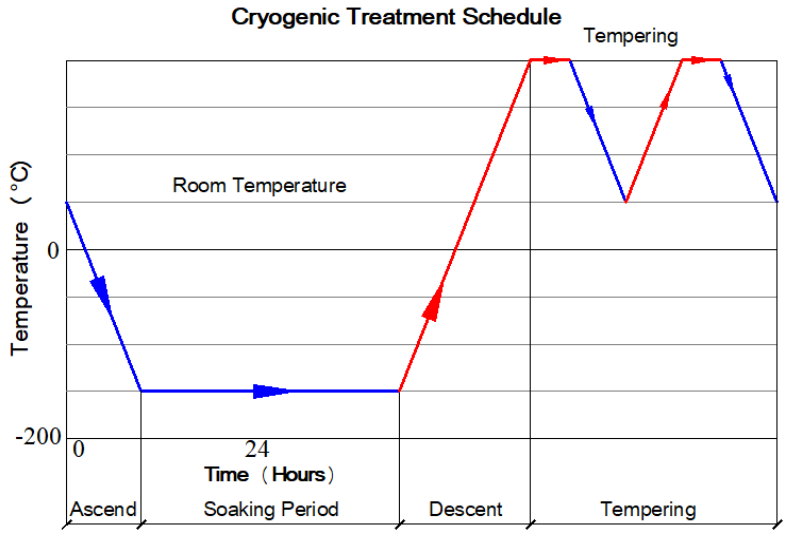


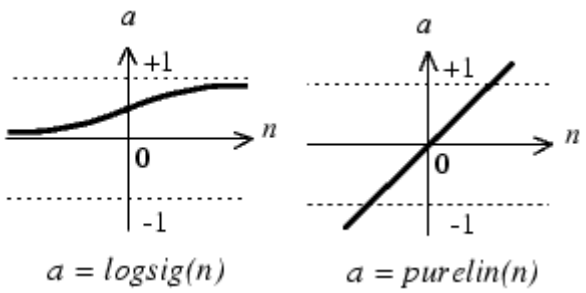
Figure 2. Deep cryogenic treatment cycle

3. ANN MODEL

Because of the fact that the ANN has a capability of solving nonlinear problems, it has been applied numerous fields for predictions used by researchers. Therefore, in present study, the ANN was employed to estimate cutting forces during turning AISI 1050 with cryo-treated and untreated CVD-coated cutting inserts. ANN methods are able to solve complex problems by using layers. Input layers were selected from cutting parameters such as cutting speed, feed rate and heat treatment condition of cutting insert while output layer was selected from cutting forces. Hidden layers received out of cutting speed, feed rate and cutting insert condition data with transfer function, which was selected as logsid transfer function, and balanced with bias weights. These transfer function types are given in Figure 3. Then, hidden layer data was transported to output layers. Finally, the result of cutting forces was computed by hidden layers and transferred to output layer by using transfer function, chosen as purelin transfer function. The created ANN model is shown schematically in Figure 4. Generally, three transfer functions were used in ANN methods. The Created ANN properties are given in Table 1.

Table 1. ANN properties

Network Type	Feed-forward back propagation
Input data	Cutting speed, feed rate, cutting insert condition
Output data	Cutting force
Training function	TRAINLM
Adaption learning function	LEARNGD
Performance function	MSE
Number of layers	3 (1 input, 1 hidden and 1 output layer)



a) b)

Figure 3. Transfer function a) Logsid b) Purelin

Depending on the input parameters, the output parameter obtained from ANN structures is given in Fig. 4.

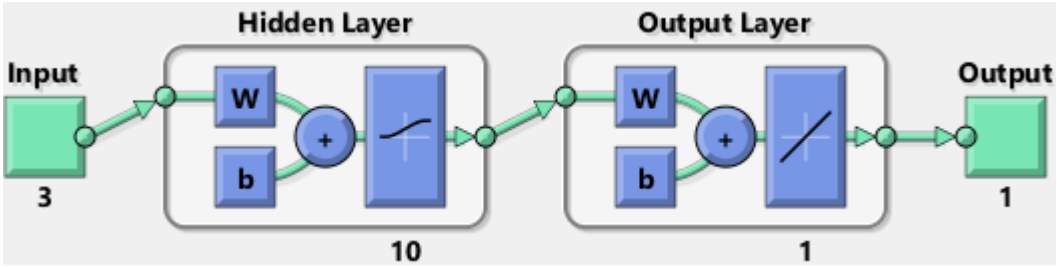


Figure 4. ANN structure

Figure 5 shows the structure of ANN used in this experimental study. Because of the number of input and output parameters, input layer has three neurons and output layer has one neuron. Hidden layers have 10 neurons in ANN structure. Depending upon the selected parameters, the ANN structure was trained, and is given in Figure. 5.

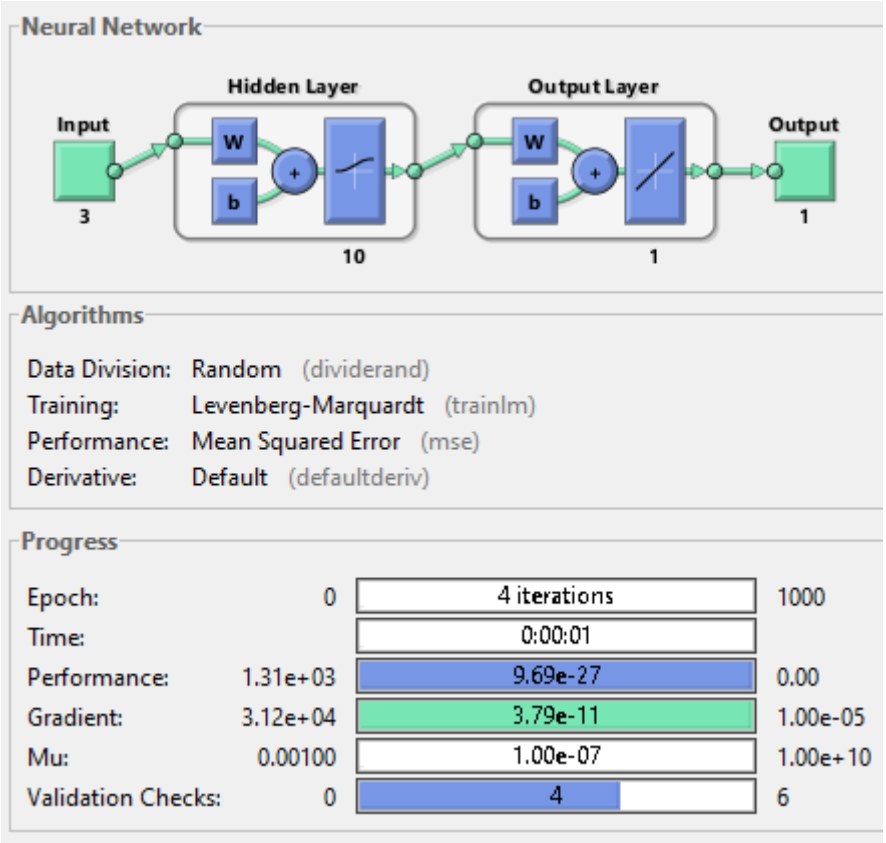


Figure 5. Neural Network Training

4. RESULTS

4.1. Results of Cutting Forces

Figure 6-7 reflect cutting forces of deep cryogenic treated inserts and untreated inserts. It is known that the cutting forces measured during machining of workpiece were increased with increasing feed rate in all cutting inserts, which is shown in Figure 6-7. According to evaluation of cutting forces, cryogenic treated cutting insert is generally lower than untreated cutting insert. This case was clearly happened at high cutting speed. This situation explains that cutting insert, which was subjected to cryogenic heat treatment, improved hardness and toughness. Furthermore, cryogenically heat treatment brings the cutting tool steel in transformation of retained and homogenously carbide distribution at subzero temperature [18, 19].

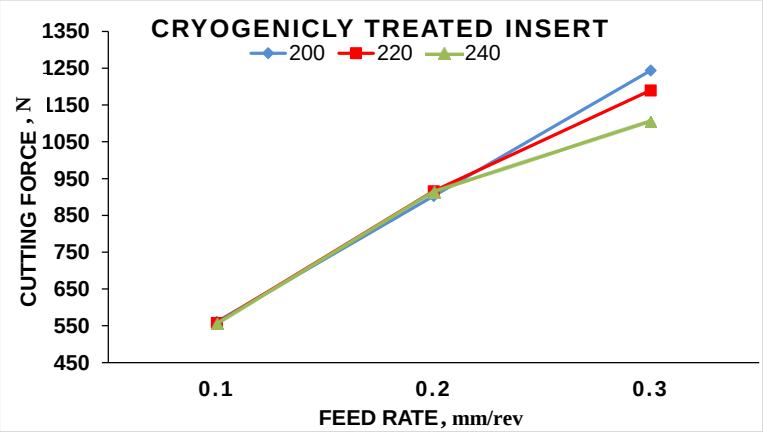


Figure 6. Cutting forces obtained from cryogenically treated insert

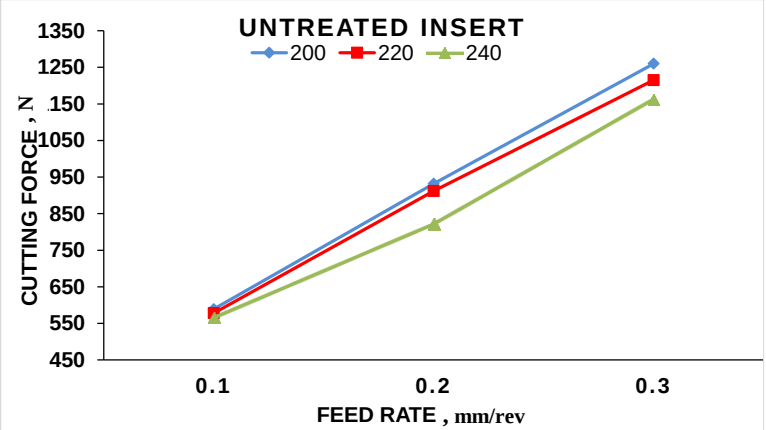


Figure 7. Cutting forces obtained from untreated insert

4.1. Results of ANN

The created ANN structure for predicting cutting forces was trained by Levenberg-Marquardt and evaluated by MSE (Mean Square Error). The result of the ANN structure in the study is shown in Figure 8, 9 and 10.

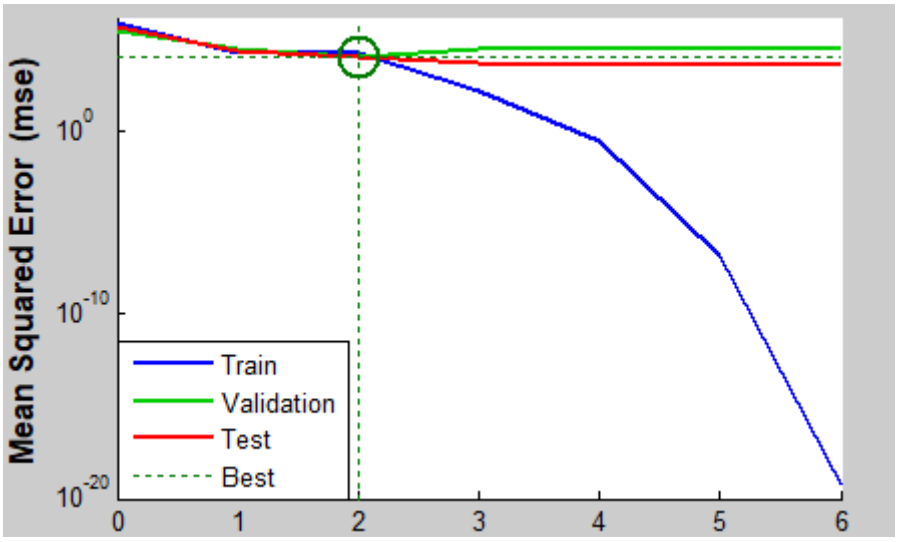


Figure 8. Neural Network Training Performance

Figure 8 shows the performance of ANN structure. The best validation performance is 11752.8556 at epoch 2 in cycling 6 epochs. In prediction capacity of ANN structure, error capacity over passing 2 epochs decreased so that the program ended in 2 epochs. Figure 9 depicts neural network training state for ANN. It shows values of gradient, mu and failure at epoch 6 ANN structure checked for validation at epoch 6. When epoch passed 1, the ANN structure stopped after validation 3.

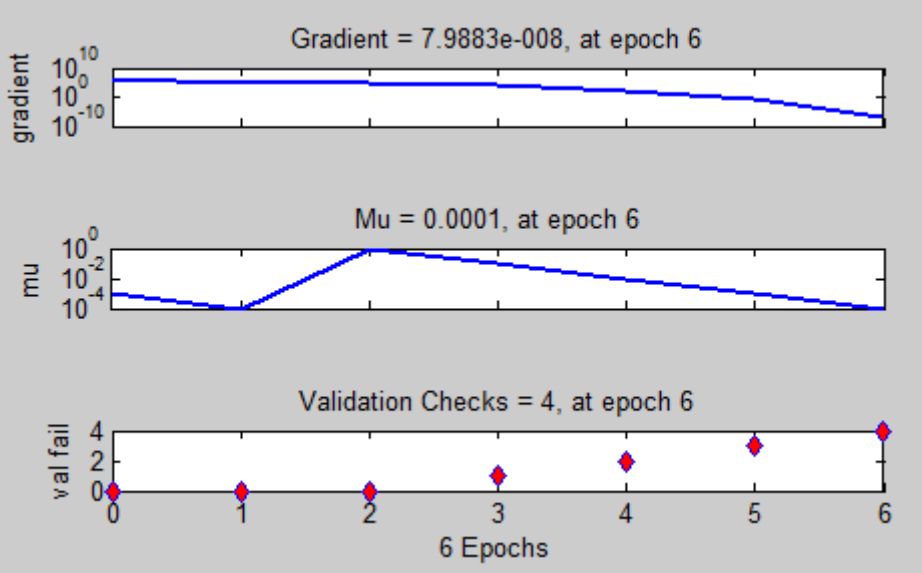


Figure 9. Neural Network Training State

The structure of trained ANN results is presented in Figure 10. It can be clearly seen from Figure 10 that R square of training, validation, test and all values are 0.99874, 0.98489, 0.99995 and 0.99575, respectively. It is said that the ANN structure estimation of all R square is enough to estimate the range from 0.9 to1. Consequently, the prediction values of cutting forces obtained from ANN structure during training and experimental values coincide completely with the regression lines, which make the R^2 = 0.99874 in training. Furthermore, all data were predicted in good agreement with each other. All predicted cutting forces are valid for the ANN structure.

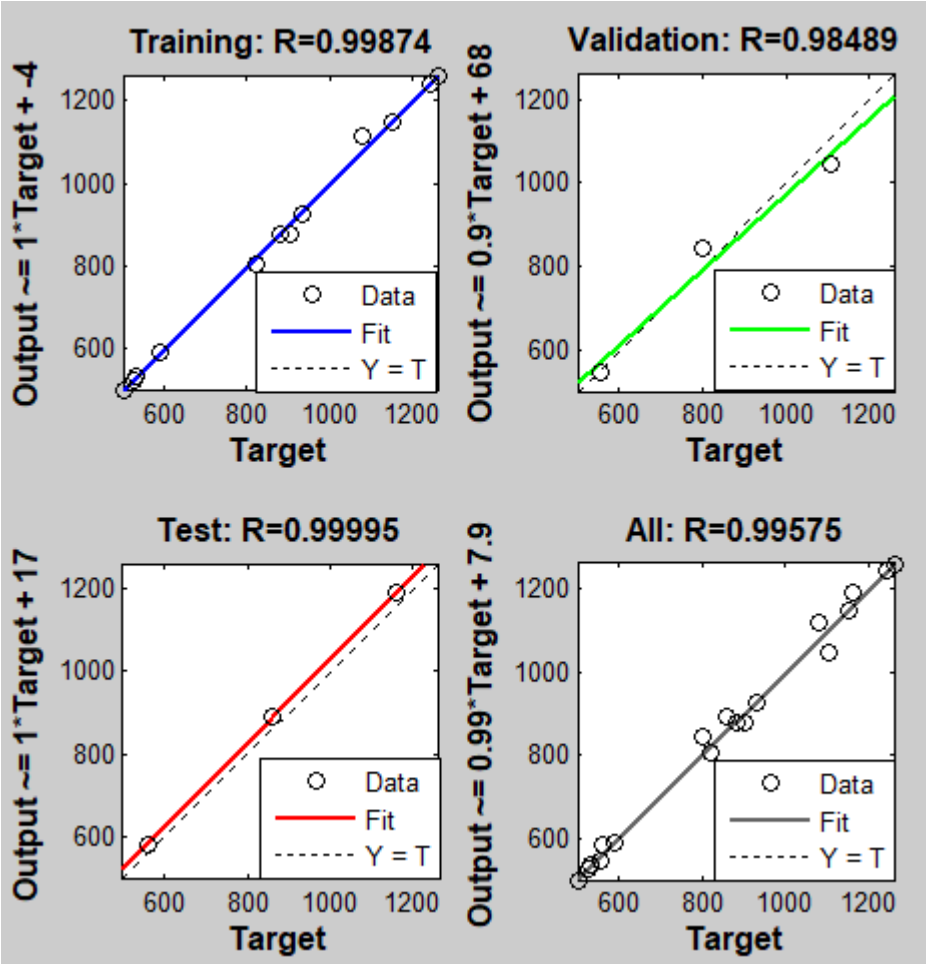


Figure 10. Neural network training regression

5. CONCLUSION

In present paper, the implementation of ANN model for estimating cutting forces was utilized with turning of AISI 1050 steel by using different cutting parameters; and the following results were drawn from this experimental study:

- The implemented ANN structure predicted outcomes of cutting forces, which were highly compatible values.
- Values of R^2 were attained from ANN structure as 0.99874, 0.98489, 0.99995 and 0.99575 for training, validation test and all, respectively.
- The experimental and prediction of cutting forces with ANN were highly consistent with each other.
- When the cutting speed was increased, cutting force decreased. However, when the feed rate was increased, cutting forces increase.

Acknowledgement

This research was supported by Batman University Scientific Research Projects Unit (BTUBAP), project no BTÜBAP-2019-YL-07. Many thanks to BTUBAP for financial support.

REFERENCES

[1] J. Kratochvíl, J. Petrů, M. Pagáč, J. Holubják, and J. Mrazik, "Effect of Chip Breakers on The Cutting Force During The Machining of Steel C45." *Advances in Science and Technology Research Journal*, vol. 11, no. 1, pp. 173-178, 2017.

[2] B. Yılmaz, Ş. Karabulut, and A. Güllü, "Performance analysis of new external chip breaker for efficient machining of Inconel 718 and optimization of the cutting parameters." *Journal of Manufacturing Processes*, vol. 32, pp. 553-563, 2018.

[3] Ş. Baday, H. Başak, and A. Güral, "Analysis of spheroidized AISI 1050 steel in terms of cutting forces and surface quality." *Kovove Mater.*, vol. 54, pp. 315-320, 2016.

[4] Ş. Baday, "Küreselleştirme ısıl işlemi uygulanmış AISI 1050 çeliğin tornalanmasında esas kesme kuvvetlerinin yapay sinir ağları ile modellenmesi." *Technological Applied Sciences*, vol. 11, no. 1, pp. 1-9, 2016.

[5] M. Hanief, , M.F. Waniand, and M.S. Charoo, "Modeling and prediction of cutting forces during the turning of red brass (C23000) using ANN and regression analysis." *Engineering science and technology, an international journal*, vol. 20, no. 3, pp. 1220-1226, 2017.

[6] H. Gürbüz, F. Sönmez, Ş. BADAY, and U. Şeker, "Farklı Talaş Kırıcı Formlarının Esas Kesme Kuvvetlerine Etkisinin Matematiksel Modellenmesi." *Batman Üniversitesi Yaşam Bilimleri Dergisi*, vol. 8, no. 2/2, pp. 13-21, 2018.

[7] H. Başak, ve Ş. Baday, "Küreselleştirilmiş orta karbonlu bir çeliğin işlenmesinde, kesme parametrelerinin kesme kuvvetleri ve yüzey pürüzlülüğüne etkilerinin regresyon analizi ile modellenmesi." *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi*, vol. 22, no 4, pp. 253-258, 2016.

[8] S. Yagmur, A. Kurt, and U. Seker, "Evaluation and mathematical modeling of delamination and cutting forces in milling of carbon fiber reinforced composite (CFRP) materials." *Journal of the Faculty of Engineering and Architecture of Gazi University*, vol. 35, no. 1, pp. 457-465, 2020.

[9] H.B. Ulas, and M.T. Ozkan, "Turning processes investigation of materials austenitic, martensitic and duplex stainless steels and prediction of cutting forces using artificial neural network (ANN) techniques." *Indian Journal of Engineering and Materials Sciences*, vol. 26, no. 2, pp. 93-104, 2019.

[10] G. Uzun, and İ. Çiftçi, "Ç 5140 çeliğinin mekanik özelliklerinin takım aşınması ve kesme kuvvetlerine etkisinin incelenmesi." Politeknik Dergisi, vol. 15, no. 1, pp. 29-34, 2012.

[11] A.İ. Özkan, İ. Sarıtaş, and S. Yaldız, "Tornalama İşleminde Kesme Kuvvetlerinin ve Takım Ucu Sıcaklığının Yapay Sinir Ağı ile Tahmin Edilmesi, *5. Uluslararası İleri Teknolojiler Sempozyumu (IATS'09)*, Karabük, Türkiye, 2009, pp. 13-15.

[12] A. Kurt, S. Sürücüler, ve A. Kirik, "Kesme Kuvvetlerinin Tahmini İçin Matematiksel Bir Model Geliştirme." *Politeknik Dergisi*, vol. 13, no. 1, pp. 15-20, 2010.

[13] S. Jeyakumar, K. Marimuthu, and T. Ramachandran, "Prediction of cutting force, tool wear and surface roughness of Al6061/SiC composite for end milling operations using RSM." *Journal of Mechanical Science and Technology*, vol. 27, no. 9, pp. 2813-2822, 2013.

[14] F. Kara, K. Aslantas, and A. Çiçek, "ANN and multiple regression method-based modelling of cutting forces in orthogonal machining of AISI 316L stainless steel." *Neural Computing and Applications*, vol. 26, no 1, pp. 237-250, 2015.

[15] I. Asilturk, H. Kahramanli, H.E. Mounayri, "Prediction of cutting forces and surface roughness using artificial neural network (ANN) and support vector regression (SVR) in turning 4140 steel." *Materials Science and Technology*, vol. 28, no. 8, pp. 980-986, 2012.

[16] N.A. Özbek, A. Çiçek, M. Gülesin, and O Özbek, "Investigation of the effects of cryogenic treatment applied at different holding times to cemented carbide inserts on tool wear." *International Journal of Machine Tools and Manufacture*, vol. 86, pp. 34-43, 2014.

[17] A.D. Shirbhate, N.V. Deshpande, and Y.M. Puri, "Effect of cryogenic treatment on cutting torque and surface finish in drilling operation with AISI M2 high speed steel." *International Journal of Mechanical Engineering Robotics Research*, vol. 1, no. 2, pp. 50-58, 2012.

[18] D. Candane, N. Alagumurthi, and K. Palaniradja, "Effect of cryogenic treatment on microstructure and wear characteristics of AISI M35 HSS." *International Journal of Materials Science and Applications*, vol 2, no. 2, pp. 56–65, 2013.

[19] S. Akincioğlu, H. Gökkaya, and İ. Uygur, "A review of cryogenic treatment on cutting tools." *The International Journal of Advanced Manufacturing Technology*, vol. 78, no 9-12, pp. 1609-1627, 2015.

Journal of Soft Computing and Artificial Intelligence
JSCAI, 1(2): pp:69-77 (2020), www.dergipark.org.tr/en/pub/jscai
JSCAI
ISSN: 2717-8226
Received: August 24 2020 Accepted: September 24 2020

RRT- Dijkstra: An Improved Path Planning Algorithm for Mobile Robots

[1]Mahmut Dirik , [2]A. Fatih Kocamaz

[1]Department of Computer Engineering, Sırnak University, TURKEY
[2]Department of Computer Engineering, Inonu University, TURKEY

Corresponding author: Mahmut Dirik (mhmd.dirik@gmail.com)

ABSTRACT The Path planning problem is one of the most researched topics in autonomous vehicles. During the last decade, sampling-based algorithms for path planning have gained significant attention from the research community. Rapidly exploring Random Tree (RRT) is a sampling-based planning approach, which is a concern to researchers due to its asymptotic optimality. However, the use of samples close to obstacles in path planning and the path with sharp turns does not make it efficient for real-time path tracking applications. For overcoming these limitations, this paper proposes a combination of RRT and Dijkstra algorithms. The RRT-Dijkstra reliefs a shorter and collision-free path solution. It is measured by various factors such as path length, execution time, and the total number of turns. The aim here is review and performance comparison of these planners based on metrics, i.e., path length, execution time, and the total number of turning points. The algorithms are tested in complex environments structured with obstacles. The experimental performance shows that RRT-Dijkstra requires less turning point and execution time in 2D environments. These are advantages of the proposed method. The proposed method is suitable for off-line path planning and path-following.

KEYWORDS: Optimal criteria, Path planning, RRT, Dijkstra, Sampling-based algorithms.

1. INTRODUCTION

Mobile robots have been effectively adapted to perform a number of tasks in various fields over the past decade. Mobile robots gain more intelligence with machine learning technologies so they can increase productivity with their navigation features. Path planning is the basis of navigation that consist of a collision-free feasible path.

Several algorithms have been used for mobile robot path planning. These are A* [1] which is an extension of the Dijkstra algorithm [2-4], Rapidly Exploring Random Tree (RRT) [5-7], Artificial Potential Fields (APF) [8, 9], visibility graph [10, 11], grid based methods [12, 13], Bidirectional-RRT (BRRT) [14, 15], Probabilistic Road Map (PRM) [16], Genetic algorithm (GA) [17-19], Fuzzy Type 1 and Fuzzy Type 2 [20, 21] planning algorithms. In global and local path planning, real-time applications, and solutions of high-dimensional problems, each of them has its own advantages and disadvantages. Several path planners can be used together to obtain a numerical value that represents the expected performance from path planning methods and techniques. The focus in this study is not only to provide a standard path planning test procedure but also to propose a solution to the path planning problem that an autonomous robot can follow in real-time in a low-cost environment. Here, RRT and Dijkstra, one of the best known/used algorithms, are examined, implemented, and tested.

The Dijkstra algorithm [2], which searches for the destination in all directions and searches for the most suitable route, is one of the most used routes planning algorithms. Using the grid map, the path

created with Dijkstra between the start and the target finds the optimum path where all the corners of the neighboring nodes are known as the edge of the straight connected link. The specific visibility points (collision-free points) are added to the map to construct the path. The vertices of the graph represent the set of configurations that are used by the planner as waypoints [22]. Dijkstra's algorithm is used in almost all path planning studies in recent years. This is because this algorithm is defined in the literature as a quick and simple route planning method [23].

RRT is a path planning techniques based on growing a random tree to search for the target position. This technique is suitable for identifying paths in uncomplicated environments without hitting obstacles to the desired target location regardless of path characteristics (ie shortest, softest etc.). RRT performs the starting path quickly and improves the quality of consecutive iterations. Thanks to its asymptotic optimal feature, it has become suitable for local path planning applications. Although the RRT algorithm is successful in navigation, convergence and real-time global path planning can also make it difficult for the mobile robot to track. This is because it has many turns and nodes close to obstacles. As a solution to this problem, the path coordinates obtained with RRT were chosen as the search area of the Dijkstra algorithm and a new path was created. Thus, the cost of the path is decreased and real-time path tracking is simplified by reducing the number of turning points.

This paper proposes an improvement over the algorithm (RRT, Dijkstra), for the optimal path planning for wheeled mobile robots. The proposed algorithm generates a better solution and shorter path with low cost. While reducing the cost of the path and reducing the turning points is the advantage of the algorithm, the increase in the computational complexity is the disadvantage. Comparative results are carried out in seven environments that confirm the effectiveness of the proposed methods. The purpose of this paper is to perform a comparative analysis of RRT and RRT+ Dijkstra based planning algorithms in a 2D environment as per stated metrics.

The rest of the paper is as follows: Section II addresses the problem methodology and explains the proposed algorithm. Section III presents the simulation results and analysis of performance. Section IV includes conclusion and future recommendations.

2. METHODOLOGY

The proposed methods were conducted in a dispersed environment with different forms of obstacles. The experimental environments used here, and the image processing processes applied to achieve these environments, were detailed in our previous study [24]. These are an extended version of the work presented in [24] with a detailed evaluation of performance parameters for RRT and Dijkstra. MATLAB and LabVIEW software were used to perform numerical and graphical for comparison and analysis. Under the same conditions, the execution time, path length, and the number of turns performance parameters have compared and tested. As input to the respective planner, 2D environment maps of different cases are provided throughout tests. The proposed algorithms are described in the following subsections.

A. RRT ALGORITHM

A rapidly exploring random tree (RRT) is a search algorithm with a single-query tree structure based on uniform random sampling, which is the start and goal point are certain known. The RRT algorithm grows based on the configurations of construction, a tree seeking the target point from a starting point, where each node of the tree is a point (state) in the workspace [25]. This algorithm was introduced by LaValle [26]. The RRT method first starts a roadmap configuration with a starting point (q_{start}) as a tree root [27]. Next, the algorithm selects a random point (q_{rand}) for each next iteration in the configuration space, and the nearest node (q_{near}) from the existing graph is searched. A new sample (q_{new}) is generated with a pre-defined distance (ε), namely a distance of step size, from q_{near} to q_{rand}. A collision is considered in the newly selected node (q_{new}). If a collision occurs, a new step (q_{new}) is discarded. Otherwise, it is added to the search tree so that for each new point, the distance between the newly generated node and the target point is considered [28]. The procedure is illustrated in Figure 1.

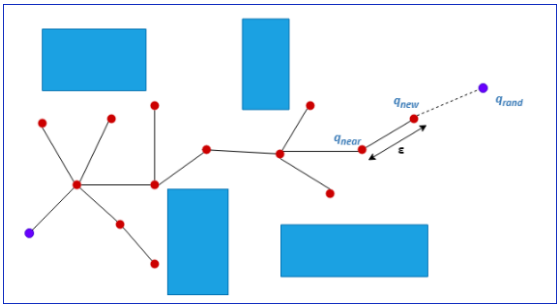


Figure 1. Extension of the RRT graph using the straight-line local planner with a resolution (ε).

Due to the random growth of the RRT, it is very time consuming when searching in the wider configuration area. The search for the growth of the tree will continue according to the predetermined threshold. The path is found if the distance is less than or equal to the set threshold. Otherwise, the distance is repeated until the distance is less than or equal to the specified threshold value, or the call times exceed the number of iterations. It can be considered using the forward tree expansion motion model, which allows the RRT to find viable trajectories even under differential constraints. That is one of the advantages and reasons for the extensive use of RRT in robotics for motion planning of systems such as mobile robots [29].

B. DIJKSTRA ALGORITHM

Dijkstra's algorithm was published by Edsger W. Dijkstra in 1959 [2]. It is a graph search algorithm for finding the shortest path between nodes with a positive edge. It is separated from the A* algorithm by not using the heuristic function [30]. Starting the calculation from the root node, the route with the lowest cost is selected, excluding routes with high-costs. The Dijkstra algorithm uses "infinite" for nodes that are marked as unvisited, and the starting point is marked as "0" (q_{init}). From the current node, the temporary distance is calculated to all the unvisited neighbors. If the newly calculated distance is shorter than the previously saved temporary distance, this is selected as a new

node. By linking the shorter distances between the nodes, the most cost-effective path in the search area given between the start and the target is determined. The algorithm's pseudo-code is given below.

Dijkstra (Graph G, Vertex s)

1. *Initialize (G,S);*
2. *Priority_Queue minQ={all vertices in V };*
3. ***while** (minQ ≠ Ø) **do***
4. *Vertex u=ExtractMin(minQ);//minimum est(u);*
5. ***for** (each v ϵ minQ such that (u,v) ϵ E*
6. *Relax (u,v);*
7. ***end for***
8. ***end while***

The algorithm maintains a priority queue minQ order that used to store the shortest path estimates (v) as key values for unvisited nodes and repeatedly extracts the minimum est(u) from minQ and relaxes all edges incident from u to any vertex in minQ.

3. PERFORMANCE ANALYSIS

A. DATA SET

The experiments have been conducted in seven configuration spaces represented by M1to M7 as shown in Figure 2. M1, M2, M4, M5 and M6 represents a basic environment with static obstacles. However, another scenario of the complex concave environments for testing the robustness of the proposed approach and safe passage is also represented by M3 and M7 maps, respectively. These are convex hull [31] technique applied environments. These environments are adopted from [24]. The proposed method is evaluated graphically and numerical comparative analysis are implemented and tested in LabVIEW+ MATLAB. The criteria to be considered when determining the optimal path for a mobile robot are measured by various factors such as path length, collision-free area, the total number of turns, and execution time. The experiments have been performed in cluttered and complex unstructured environments with obstacles of different shapes. The results of all algorithms provided in same place for comparing performance parameters in path length, execution time, and the number of graphs.

B. EXPERIMENTAL RESULTS

In this section, we present the experimental results of the proposed algorithm and the comparison of their results. All approaches are shown for maps M1 to M7 in Figure 2. Path length, execution time, total number of nodes used in the path grid with comparison are also shown in Table 1 and Table 2. The visual comparison of RRT and RRT+ Dijkstra algorithms are shown in Fig. 3 (a) and (b). From these Figures it is clear that the proposed approaches has generated shorter paths and less turning points than RRT methods. It was seen that the execution time of the algorithm increased. This is because the two algorithms worked consecutively. The node-set (number of cells) generated by

RRT algorithm created the search space of the Dijkstra algorithm. The proposed approach is flexible enough to generate safe and shorter path with less turning points. Reducing the number of nodes processed also reduced the path length by eliminating extra path points on the path.

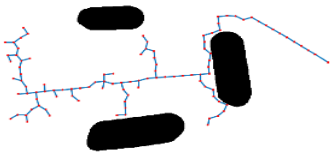
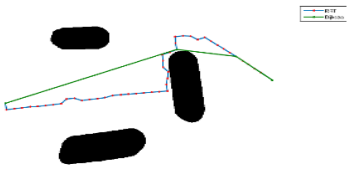
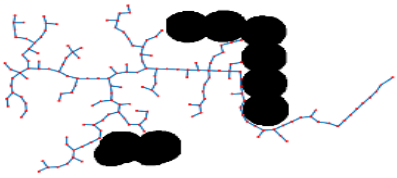
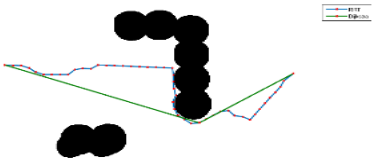
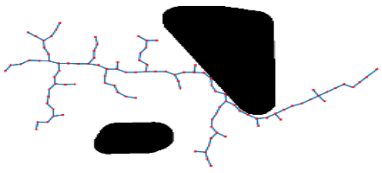
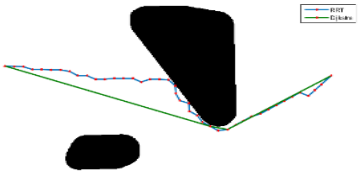
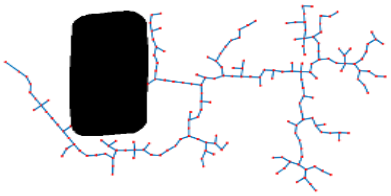
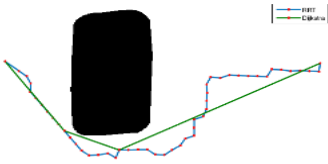
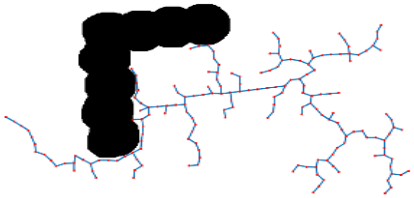
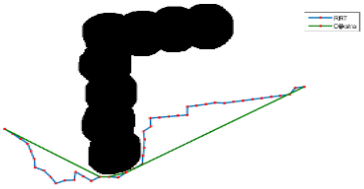
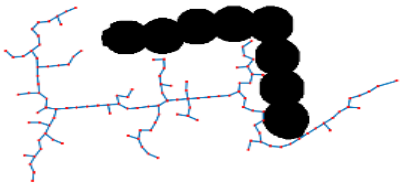
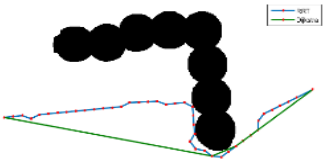
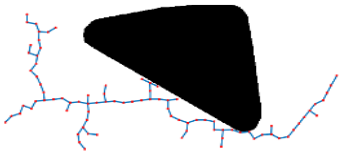
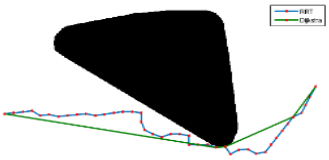
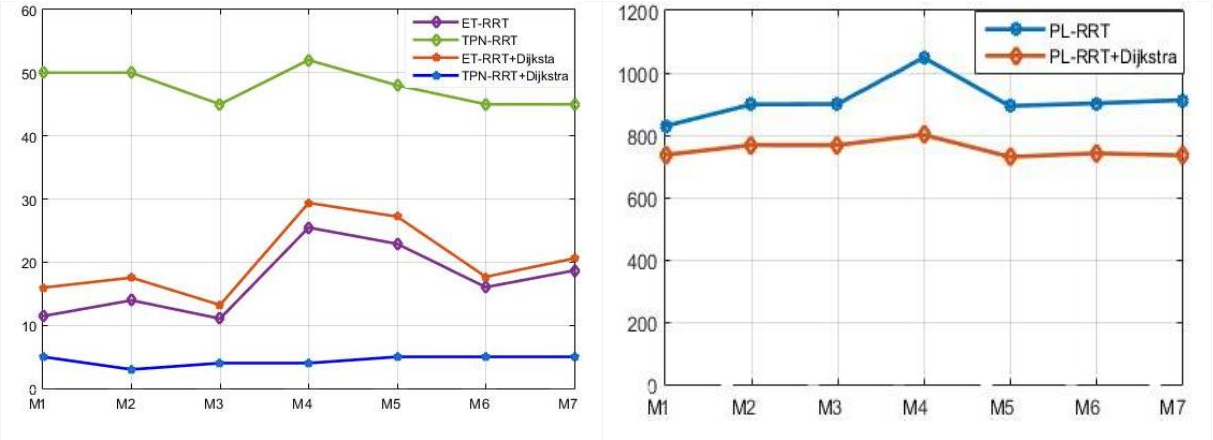
Exp.	RRT Tree	RRT+ Dijkstra Path
M1		
M2		
M3		
M4		
M5		
M6		
M7		

Figure 2. Experimental results of environment maps M1 to M7 using RRT and RRT+ Dijkstra

The path created with RRT is not possible to follow because of the dynamic and kinematic constraints of mobile robots. This requires the robot to perform complex movements to track sharp

turns, resulting in high energy consumption, controller use, and premature aging of the robotic parts. The proposed method is presented as a solution to such situations in global path planning



a) b)

Figure 3. Plot of path length (b), execution time and total path nodes (a) of all planners for all environment maps (PL: Path Length, ET: Execution Time, TPN: Total Path Nodes).

Plots of experimental results of the all planners are shown in Fig. 3. It is obvious from the results that the proposed methods generate shorter and relatively better solution paths for the densely cluttered environments. The proposed methods require fewer nodes, which makes the algorithm more efficient. From comparisons with RRT and RRT+ Dijkstra, we draw a conclusion that the performance of proposed methods in PL and TPN is best overall. However, the proposed algorithm had significantly lower performance, mainly due to the high computational time. For smoother routes and greater security levels, the proposed methods can be selected. It is suitable for wheeled mobile robot applications in offline path tracking. The numerical equivalent of the values of the above graph is given in Table 1.

Table 1. Experiments result for all planners (**PL:** Path Length, **ET:** Execution Time, **TPN:** Total Path Nodes)

Algorithm		**M1**	**M2**	**M3**	**M4**	**M5**	**M6**	**M7**
RRT	**PL-(px)**	829,24	898,68	899,95	1048,7	893,76	902,20	912,45
	ET-(sc)	11,45	13,95	11,05	25,47	22,90	16,03	18,67
	TPN-(nodes)	50	50	45	52	48	45	45
	Z-Score	0,554	-0,155	0,257	0,454	0,168	0,327	-0,119
RRT+Dijkstra	**PL-(px)**	736,78	768,07	767,99	801,29	730,30	741,68	735,05
	ET-(sc)	15,95	17,54	13,21	29,38	27,21	17,67	20,59
	TPN-(nodes)	55	53	49	56	53	49	50
	Z-Score	0,127	-0,818	-0,063	0,302	-0,015	0,179	-0,263

Table 2 presents statistical analysis of results. PL, ET and TPN in all environments maps for all planner are listed in this Table. RRT+ Dijkstra algorithms generate better path with less PL and TPN.

	M1	**M2**	**M3**	**M4**	**M5**	**M6**	**M7**

PL-(px)	11,15 % less	14,53 % less	14,66 % less	23,59 % less	18,29 % less	17,79 % less	19,44 % less
ET-(sc)	39,30 % more	25,73 % more	19,55 % more	15,35 % more	18,82 % more	10,23 % more	10,28 % more
TPN-(nodes)	90,00 % less	94,00 % less	91,11 % less	92,31 % less	89,58 % less	91,11 % less	88,89% less

Table 2. Statistically, comparison of all approaches for ET, PL, and TPN (PL: Path Length, ET: Execution Time, TPN: Total Path Nodes

Both algorithms (RRT and RRT+ Dijkstra) are efficient at finding an obstacle-free path in different experimental environments. However, it has been observed that the proposed planner reduces PL and TPN compared to traditional RRT and also increases ET. It is made possible by the proposed method to obtain values that can further reduce the path length and the number of path nodes and result in a flattened path. It is relatively better solution.

4. CONCLUSIONS

In this paper, image-based RRT and Dijkstra algorithms are presented to find collision-free path planning in a cluttered environment. The performance of RRT was compared with the algorithms of RRT + Dijkstra for different scenarios.

The experimental results show that the path obtained only using RRT is not suitable for the mobile robot to follow because it has sharp transitions and a curved structure. Considering the dynamic and kinematic constraints of mobile robots, it is desirable to develop an appropriate algorithm for global path planning and path tracking. Because performing complex movements to follow sharp turns can result in the robot's high energy consumption, controller use, and premature aging of the robotic parts.

In order to improve this situation or obtain more suitable path coordinates, RRT and Dijkstra algorithm have been applied together. It is clear that due to the use of Dijkstra and RRT together, the total length of the path decreases while the time spent in path planning is prolonged. The method we propose is mainly aimed at creating a costly and traceable route. It has been proven that RRT + Dijkstra effectively produces shorter paths. In addition, computing efficiency and less memory requirements are important criteria for small robots. Experiments are carried out in different 2D environments. The results obtained also show the effectiveness and applicability of the proposed method. In future research, we plan to carry out path tracking application of mobile robots on the planned route in real time.

REFERENCES

[1] F. Duchoň *et al.*, "Path Planning with Modified a Star Algorithm for a Mobile Robot," *Procedia Eng.*, vol. 96, pp. 59–69, 2014, doi: 10.1016/j.proeng.2014.12.098.

[2] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numer. Math.*, vol. 1, no. 1, pp. 269–271, Dec. 1959, doi: 10.1007/BF01386390.

[3] S. A. Fadzli, S. I. Abdulkadir, M. Makhtar, and A. A. Jamal, "Robotic Indoor Path Planning using Dijkstra ' s Algorithm with Multi-Layer Dictionaries," pp. 1–4, 2015.

[4] P. Hart, N. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. 4, no. 2, pp. 100–107, 1968, doi: 10.1109/TSSC.1968.300136.

[5] J. J. Kuffner and S. M. La Valle, "RRT-connect: an efficient approach to single-query path planning," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2000, doi: 10.1109/robot.2000.844730.

[6] J. Bruce and M. Veloso, "Real-time randomized path planning for robot navigation," in *IEEE/RSJ International Conference on Intelligent Robots and System*, 2002, vol. 3, pp. 2383–2388, doi: 10.1109/IRDS.2002.1041624.

[7] A. Bry and N. Roy, "Rapidly-exploring random belief trees for motion planning under uncertainty," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2011, doi: 10.1109/ICRA.2011.5980508.

[8] T. Weerakoon, K. Ishii, and A. A. F. Nassiraei, "An Artificial Potential Field Based Mobile Robot Navigation Method To Prevent From Deadlock," *J. Artif. Intell. Soft Comput. Res.*, vol. 5, no. 3, pp. 189–203, Jul. 2015, doi: 10.1515/jaiscr-2015-0028.

[9] E. Rimon and D. E. Koditschek, "Exact robot navigation using artificial potential functions," *IEEE Trans. Robot. Autom.*, vol. 8, no. 5, pp. 501–518, Oct. 1992, doi: 10.1109/70.163777.

[10] H. Kaluder, M. Brezak, and I. Petrović, "A visibility graph based method for path planning in dynamic environments," in *MIPRO 2011 - 34th International Convention on Information and Communication Technology, Electronics and Microelectronics - Proceedings*, 2011.

[11] R. Wein, J. P. Van Den Berg, and D. Halperin, "The visibility-Voronoi complex and its applications," in *Computational Geometry: Theory and Applications*, 2007, doi: 10.1016/j.comgeo.2005.11.007.

[12] P. Yap, "Grid-based path-finding," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2002, doi: 10.1007/3-540-47922-8_4.

[13] A. I. Panov, K. S. Yakovlev, and R. Suvorov, "Grid path planning with deep reinforcement learning: Preliminary results," in *Procedia Computer Science*, 2018, doi: 10.1016/j.procs.2018.01.054.

[14] E. Donmez, A. F. Kocamaz, and M. Dirik, "Bi-RRT path extraction and curve fitting smooth with visual based configuration space mapping," in *International Artificial Intelligence and Data Processing Symposium (IDAP)*, Sep. 2017, pp. 1–5, doi: 10.1109/IDAP.2017.8090214.

[15] R. Sadeghian, S. Shahin, and M. T. Masouleh, "An experimental study on vision based controlling of a spherical rolling robot," in *Iranian Conference on Intelligent Systems and Signal Processing (ICSPIS)*, Dec. 2017, pp. 23–27, doi: 10.1109/ICSPIS.2017.8311583.

[16] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Autom.*, vol. 12, no. 4, pp. 566–580, 1996, doi: 10.1109/70.508439.

[17] C. Lamini, S. Benhlima, and A. Elbekri, "Genetic Algorithm Based Approach for Autonomous Mobile Robot Path Planning," *Procedia Comput. Sci.*, vol. 127, pp. 180–189, 2018, doi: 10.1016/j.procs.2018.01.113.

[18] Jianping Tu and S. X. Yang, "Genetic algorithm based path planning for a mobile robot," in *International Conference on Robotics and Automation (Cat. No.03CH37422)*, 2003, vol. 1, pp. 1221–1226, doi: 10.1109/ROBOT.2003.1241759.

[19] L. Moreno, J. M. Armingol, S. Garrido, A. De La Escalera, and M. A. Salichs, "A genetic algorithm for

mobile robot localization using ultrasonic sensors," *J. Intell. Robot. Syst. Theory Appl.*, 2002, doi: 10.1023/A:1015664517164.

[20] J.-Y. Jhang, C.-J. Lin, C.-T. Lin, and K.-Y. Young, "Navigation Control of Mobile Robots Using an Interval Type-2 Fuzzy Controller Based on Dynamic-group Particle Swarm Optimization," *Int. J. Control. Autom. Syst.*, vol. 16, no. 5, pp. 2446–2457, Oct. 2018, doi: 10.1007/s12555-017-0156-5.

[21] T. W. Liao, "A procedure for the generation of interval type-2 membership functions from data," *Appl. Soft Comput.*, vol. 52, pp. 925–936, Mar. 2017, doi: 10.1016/j.asoc.2016.09.034.

[22] M. Dirik, "Development of vision-based mobile robot control and path planning algorithms in obstacled environments," Inonu University, 2020.

[23] M. Fu, J. Li, and Z. Deng, "A practical route planning algorithm for vehicle navigation system," in *Proceedings of the World Congress on Intelligent Control and Automation (WCICA)*, 2004, doi: 10.1109/wcica.2004.1343742.

[24] M. Dirik and A. F. Kocamaz, "Global Vision Based Path Planning for AVGs Using A* Algorithm." Journal of Soft Computing and Artificial Intelligence, 1, pp. 18–28, 2020.

[25] X. Cao, X. Zou, C. Jia, M. Chen, and Z. Zeng, "RRT-based path planning for an intelligent litchi-picking manipulator," *Comput. Electron. Agric.*, vol. 156, pp. 105–118, Jan. 2019, doi: 10.1016/j.compag.2018.10.031.

[26] S. M. LaValle, "Rapidly-Exploring Random Trees: A New Tool for Path Planning," *Iowa State Univ. Ames, IA 50011 USA*, vol. 6, no. 2, p. 103, 1998.

[27] N. Chao, Y. Liu, H. Xia, A. Ayodeji, and L. Bai, "Grid-based RRT∗ for minimum dose walking path-planning in complex radioactive environments," *Ann. Nucl. Energy*, vol. 115, pp. 73–82, May 2018, doi: 10.1016/j.anucene.2018.01.007.

[28] L. Jaillet, A. Yershova, S. M. La Valle, and T. Simeon, "Adaptive tuning of the sampling domain for dynamic-domain RRTs," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005, pp. 2851–2856, doi: 10.1109/IROS.2005.1545607.

[29] A. Viseras, D. Shutin, and L. Merino, "Robotic Active Information Gathering for Spatial Field Reconstruction with Rapidly-Exploring Random Trees and Online Learning of Gaussian Processes," *Jr. Sensors*, vol. 19, no. 5, pp. 10–16, Feb. 2019, doi: 10.3390/s19051016.

[30] G. Klančar, A. Zdešar, S. Blažič, and I. Škrjanc, *Wheeled Mobile Robotics, From Fundamentals Towards Autonomous Systems*. Butterworth-Heinemann, © 2017 Elsevier Inc., 2017.

[31] F. Yaacoub, Y. Hamam, A. Abche, and C. Fares, "Convex Hull in Medical Simulations: A New Hybrid Approach," in *IECON 2006 - 32nd Annual Conference on IEEE Industrial Electronics*, Nov. 2006, pp. 3308–3313, doi: 10.1109/IECON.2006.347668.

Journal of Soft Computing and Artificial Intelligence
JSCAI, 1(2): pp:78-85 (2020), www.dergipark.org.tr/en/pub/jscai
JSCAI
ISSN: 2717-8226
Received: September 22 2020 Accepted: October 9 2020

RGBSticks: A New Deep Learning Based Framework for Stock Market Analysis and Prediction

[1]Eren Unlu

[1]Supelec, Paris, 75000 FR

Corresponding author: Eren Unlu (e-mail: datascientist.unlu@ gmail.com).

ABSTRACT We present a novel intuitive graphical representation for daily stock prices, which we refer to as RGBSticks, a variation of classical candle sticks. This representation allows the usage of complex deep learning based techniques, such as deep convolutional auto encoders and deep convolutional generative adversarial networks to produce insightful visualizations for market's past and future states. We believe RGBStick representation has great potential to integrate human decision process and deep learning for stock market analysis and forecasting. The traders who are highly familiar with candlesticks are able to evaluate the results generated by deep learning algorithms by inspecting the varying color shades in a compact, instinctual and rapid fashion.

KEYWORDS: deep learning, forecasting, time series.

1. INTRODUCTION

Candlestick charts have been by far the most widespread way of visualizing the daily stock market and exchangeable currency, derivative or commodity values for a long time. According to many sources, they can be traced back to Japan's Meiji period in the 18th century, developed for rice trading. Some academics particularly credit it to Munehisa Homma, a prominent rice broker of the era [1, 2]. The very fact that they have been used uninterruptedly and unchanged throughout the history of modern trading hints us about the effectiveness of the method. Even with today's complex cutting edge digital infrastructure and automated trading algorithms; they are still among the most valuable elements of a broker's toolbox; and this is likely to last for years. Possibly, it's effectiveness comes from its simplicity, where the overall image of many consecutive days (or any other intended time period) can abstract many hidden and latent factors of market dynamics inside a broker's mind. The human brain's preference for visual data to rapidly process complex tasks subconsciously is a well-studied phenomenon [3, 4].

Based on this observation, we propose to process the candlestick charts as an image, in contrast to many deep learning and data science based stock price analysis and forecasting techniques, which approaches the issue as raw tabular or temporal feature extraction [5-8].

The proposed method's advantage appears to be twofold. Firstly, it may allow the deep learning algorithms to capture the complex patterns that raw data processing cannot provide; an insight stemming from the success of candlesticks that drove human traders to make correct decisions in a highly chaotic environment throughout three centuries. Second, as the results provided by the deep learning methods are also candlestick like presentations; the human traders can gain insights on the past and the future market and conclude with interpretable artificial intelligence support.

After we have come up with the idea, firstly we have reviewed the literature to check whether a similar approach was followed before. To the best of our knowledge, [9] is the only one to mention the encoding of stock price movements as candlestick images, however it still does not propose a pure visual representation like the method explained in this paper. The authors use Gramian Angular Fields to encode the stock market time series and apply deep Convolutional Neural Networks (CNN).

Candlestick approach is highly straightforward: The value of the traded entity (currency, commodity, contract etc.) in a time period (a day, a business week, 5 minutes etc.) are represented with four values. *Open* and *Close* are the first and last are the prices at the beginning and end of the period, respectively. *High* and *Low* values correspond to the highest and the lowest ever price in the particular period of interest. If the close value is lower than the open value, it is considered as a *bearish* period and the inner candle of the candlestick is colored red. The inverse case corresponds to a *bullish* period and is represented with a green candle. A large gap between high and low indicates a high volatility in that particular period.

In this paper, we present *RGBSticks*, a novel framework to transform candlestick market prices to a structure, where it is still both readable for human agents and also in the form which benefits the deep learning methods. The name comes from the encoding of open, close, low and high prices on red, green and blue channels of a digital image. After explaining this visual representation, we analyze the daily stock price of a company by using a dense deep neural network based auto encoder. By using the same architecture the next day is tried to be predicted. Finally, a Deep Convolutional Generative Adversarial Network (DC-GAN) architecture is applied to simulate artificial market periods.

2. RGBSticks

An RGBStick is visually almost identical to a conventional candlestick as it can be seen Fig. 1, with the exception that the outer sticks have the same width with the inner candle and are encoded in blue channel. This allows the deep learning based computer vision algorithms to process and output human readable three dimensional (width, height indices and 3 channel color) images. Thanks to this representation, we can proliferate the performance of deep learning algorithms which use this kind of input shape, forcing the synaptic and convolutional weights to be tuned to plausible and distinctive ranges by assigning the information to highest values of a single channel.

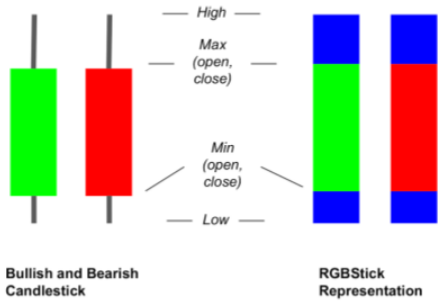


Figure 1. A traditional candlestick representation and an RGBStick representation for a bullish and bearish period.

A candlestick transforms the open, close, low, high values into 5 semantic visual features : high, low, maximum and minimum of the open and close price and whether the close value was higher than the open or not, where the last is represented with red or green color.

Temporal deep Learning algorithms, whether they process images (e.g. video frame prediction) or not require the supervision of the sequential data in to a tabular structure, where the lagged instances are grouped together. This procedure is sometimes referred to as time series supervision or lagged/sliding window feature generation. The maximum length of each lagging window is called the *look back* parameter, that being also the expression used in this paper. Similarly, the output data point length is referred to as look after, the number of instances in future we want to predict. In this paper, look back and look after values are determined as 16 and 1, meaning we want our machine learning algorithms to forecast the next day based on previous 16 days. So, each input data point in our system is a square image, as in Fig. 2.. And the output data point is a rectangular image representing a day. For each 16 days, we first calculate the maximum and minimum of all 4 indicators. Without loss of generality, we assume that the values of the next day will not be larger than the 10% of the maximum and smaller than 10% of the minimum value of the previous 16 days. Thanks to this defined global maximum and minimum, we can normalize and place RGBSticks in the vertical axis for a specific time block. In case, for the relatively rarer cases where the next day is out of these ranges its values are clipped. The image size of 64x64 pixels is determined for this paper, thus the width and height of a single RGBStick is normalized according to.

As you can see, the candlesticks are placed on a total black image plane, where all the 3 channel values have 0 intensity. As the information on being bullish or bearish and the volatility (the outer limits of a RGBStick) are encoded as the highest intensity of all independent 3 channels; this permits the machine learning algorithms to capture and exploit the most important features of the data by enforcing optimization process to iterate through the extrema.

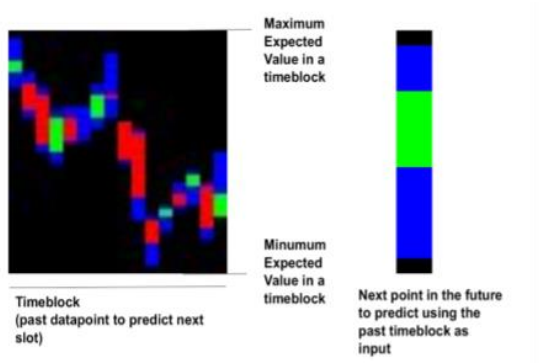


Figure 2. A time block of look back long days from the whole stock price data.

This corresponds to a single data point of the input batches both for training and testing segments. The output data point is similarly the following look after long block, which is assigned as the output. A look back value of 16 is determined for this work. For forecasting task, we have defined the look after value as one, which means that we want our deep learning model to predict the next day by evaluating the previous 16 days. In case of auto encoding and adversarial networks the output data point is the same with the input.

3. Deep Neural Network Forecasting of Future Market

Based on the explained framework, as a first attempt, we have built a regular dense deep neural network to predict the next days based on the previous 16 days. An hourglass shaped, auto encoder like architecture is preferred to reduce dimensionality on latent feature space to mitigate the highly volatile nature of stock market data. The details of the used neural network can be inspected in Fig. 3. We present the results of our work for a single company's daily stock market prices for a calendar year. The first 90% of the days are used as training, whilst the remaining last days of the years are evaluated for testing purposes. As it can be seen from the figures, our RGBStick backed deep learning framework not only provides accurate results, it also permits the human trader to develop a market strategy by evaluating the results based on the color shades, similar to traditional candlestick charts, where he/she already is familiar.

Figure 3. The hourglass shaped deep dense neural network used for predicting the next days' RGBsticks based on previous 16 days. The relu layers are used in all layers, except the last layer, where logistic regression probability is preferred for matching the output.

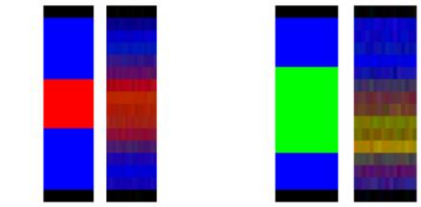


Figure 4. Real RGBStick and the predicted one with our dense deep neural network for a bearish and bullish day in the test dataset.

As it can be noted, the algorithm performs quite efficiently to predict the tendency and volatility. A human interpreter can have an insight on the volatility and the tendency of the forecast by evaluating the difference hue variations. Intensity of the blue on the outer edges signal the limits and confidence of the volatility. Similarly, stronger reddish or greenish hue give an idea on the upper and lower limits of close and open values. For instance yellowish hue would mean the mixture of green and blue; where open/close values are close to each other for a bullish day. In contrast, violet hue pixels can be interpreted for the same effect for a bearish day. Brownish pixels can be interpreted as the confidence on bearish/bullish decision is lower as mixture of red and green.

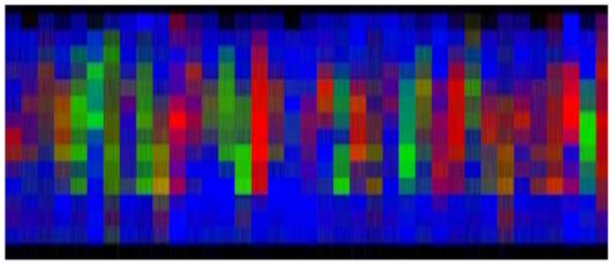


Figure 5. The predicted RGBSticks days of the test part.

This kind of concatenated view permits the human trader to evaluate the forecasts and the short to mid term tendency in the market in a global fashion with the parameters such as confidence, volatility and extrema are encoded in color shades.

4. Deep Autoencoders for Understanding Dynamics of Market

We have used the above mentioned architecture as a deep autoencoder, with the intentions to extract information out of the history of the stock prices. As the architecture was chosen primarily as an hourglass structure it suits well to the concept. As it is known, autoencoders takes the same input and output data points to be trained, thus they are classified as self-supervised. The data is projected in to a lower dimensional latent feature space, which is called as the bottleneck for extracting the most valuable information, eliminating the noisy parts.

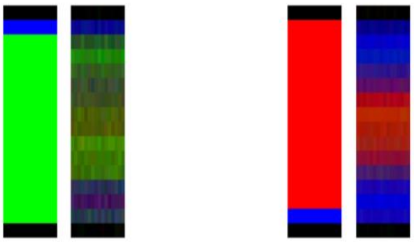


Figure 6. The days in the past of the stock prices with real and the predicted RGBSticks using deep auto encoder. The auto encoder allows us to interpret that on these specific two days open/close values would actually meant to be more closer to each other based on the overall dynamics of the environment.

5. Market Analysis and Simulation with Deep Convolutional Generative Adversarial Networks

Generative Adversarial Networks (GAN) are relatively new architectures in deep learning research, being started to be developed a few years ago [10-12]. They are one of the most innovative breakthroughs in artificial intelligence. With these state-of-the-art models we are now capable of generating highly sophisticated artificial data, such as deep fakes or artificial human faces or transfer styles between batches of images. The central idea is to have two separate neural networks connected to each other called a generative network and a discriminator network. In the case for convolutional networks for images, the generator network takes randomized inputs for its latent space and maps it to the images of the intended size through deconvolution layers. As for the variational auto encoders, GAN also learns the distribution via latent space rather than the direct processing of input data. The discriminator's task is to be fed with real training input images and the artificial outputs of the generative network, and classify it as a real one or fake one. Throughout training, generative network learns gradually to produce more real like artificial outputs, whilst discriminative network gets better to discriminate the fake ones. Thanks to this adversarial concept, at the end we are able to produce realistic fake images. Even though it is an adversarial setting, the nature is cooperative, where at the end generator network is able to generate realistic fake images. The DC-GAN architecture is highly efficient when you consider the fact that the latent vector is fed with random inputs. Adversarial setting and deep convolutional and dense layers is capable of being trained in short time to produce plausible results [12].

We have trained a deep convolutional generative adversarial network with the daily stock prices represented as RGBStick images. The details of the used architecture can be seen in Fig. 7. We provide 32 arbitrary real RGBStick time blocks of the stock price history of the company and 24 fake time blocks generated by the DC-GAN architecture in Fig. 8. Note the fact that, the deep generator has grasped the sharp upside and downside trends and high volatility in the data.

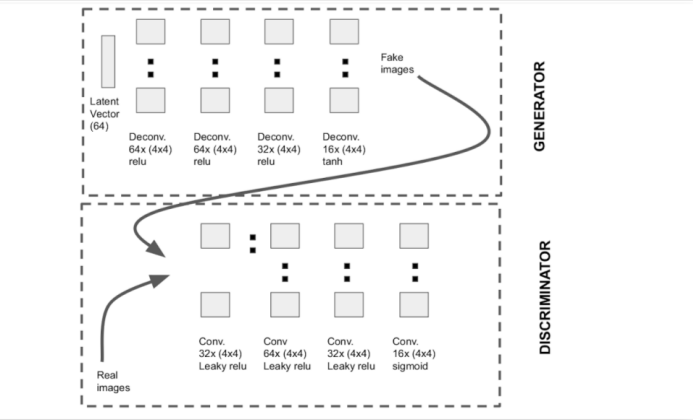


Figure 7. DC-GAN architecture used to train our RGBStick representations of stock market which allows us to simulate the dynamics of the market but also understand its underlying complex statistical structure by inspecting the variations of different random numbers on the latent vector of the generator.

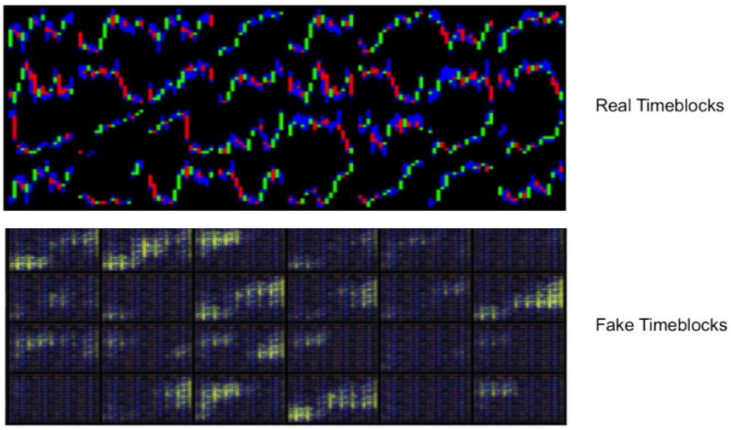


Figure 8. 32 arbitrary real RGBStick time blocks of the stock price history of the company and 24 fake time blocks generated by the DC-GAN architecture.

6. CONCLUSION

Candlestick charts has been used extensively for a long time for the analysis of trading markets. The success of this kind of a chart comes from its effective visual abstraction for human traders, who can have a wide angle view of the history. This has encouraged us to theorize that there might be a powerful latent information encoded visually. Thus, we propose to represent open, close, high and low prices as candlestick like graphical representation, which we refer as RGBSticks. The outer sticks defining the difference between the high and low values, i.e. the limits of the volatility in the time period is encoded as full intensity in the blue channel (thus, zero intensity in the other 2 color channels). The bearish and bullish inner candles are represented as full intensity in the green and red channels. This encoding of important information on the extrema of color channels helps deep learning algorithms to reach more optimal weights for intended tasks. We have tested a dense neural network to predict the RGBStick of the next day based on previous 16 days. The same architecture is used also to evaluate the history of

the stock prices by auto encoding. Finally, a DC-GAN architecture is applied to simulate and understand the stock prices of a company.

We believe RGBStick representation has great potential to integrate human decision process and deep learning for stock market analysis and forecasting. The traders who are highly familiar with candlesticks are able to evaluate the results generated by deep learning algorithms by inspecting the varying color shades in a compact, instinctual and rapid fashion.

REFERENCES

[1] G. L. Morris, Candlestick Charting Explained: Timeless Techniques for Trading Stocks and Futures: Timeless Techniques for Trading stocks and Sutures. McGraw Hill Professional, 2006.

[2] S. Nison, Japanese candlestick charting techniques: a contemporary guide to the ancient investment techniques of the Far East. Penguin, 2001.

[3] A. Berti and G. Rizzolatti, "Visual processing without awareness: Evidence from unilateral neglect," Journal of cognitive neuroscience,vol. 4, no. 4, pp. 345–351, 1992.

[4] P. Lewicki, M. Czyzewska, and H. Hoffman, "Unconscious acquisition of complex procedural knowledge." Journal of Experimental Psychology: Learning, Memory, and Cognition, vol. 13, no. 4, p. 523, 1987.

[5] E. Chong, C. Han, and F. C. Park, "Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies," Expert Systems with Applications, vol. 83, pp. 187–205, 2017.

[6] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," European Journal of Operational Research, vol. 270, no. 2, pp. 654–669, 2018.

[7] A. Yoshihara, K. Fujikawa, K. Seki, and K. Uehara, "Predicting stock market trends by recurrent deep neural networks," in Pacific rim international conference on artificial intelligence. Springer, 2014, pp. 759–769.

[8] Z. Hu, W. Liu, J. Bian, X. Liu, and T.-Y. Liu, "Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction," in Proceedings of the eleventh ACM international conference on web search and data mining, 2018, pp. 261–269.

[9] J.-H. Chen and Y.-C. Tsai, "Encoding candlesticks as images for pattern classification using convolutional neural networks," Financial Innovation, vol. 6, pp. 1–19, 2020.

[10] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in International Conference on Machine Learning. PMLR, 2019, pp. 7354–7363.

[11] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," arXiv preprint arXiv:1511.06434, 2015.

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in Advances in neural information processing systems, 2014, pp. 2672–2680.

Journal of Soft Computing and Artificial Intelligence
JSCAI, 1(2): pp:86-91 (2020), www.dergipark.org.tr/en/pub/jscai
JSCAI
ISSN: 2717-8226
Received: October 27 2020 Accepted: November 19 2020

Applying Decision Tree Techniques to Classify European Football Teams

[1]Bünyamin Fuat Yıldız

[1]M.Sc. (Econ.), 24 Crooks Ave Apt 229, Clifton, NJ 7011-1614 United States

Corresponding author: Bünyamin Fuat Yıldız e-mail: bunyaminfuatyildiz@yahoo.com

ABSTRACT Machine learning techniques are powerful tools used in all aspects of science. However, these techniques are relatively new in sports. This study was carried out to measure the accuracy of decision trees in the classification of football teams. We applied five types of decision tree algorithms to classify elite football teams in Spain, Italy, and England to determine whether decision tree techniques are robust in classifying elite football teams. The findings show that the accuracy rate is above 77 percent for each of the decision trees. The key qualities that cause branching in decision trees may constitute a criterion for the targeting of football authorities. More research is required to determine which machine learning techniques are more efficient in classifying football teams.

KEYWORDS: Machine Learning, Decision Trees, Football, Classification, Sports.

1. INTRODUCTION

The role of machine learning has received increased attention across several disciplines in the last three decades. Recently scholars have applied machine learning techniques in several sports from ice hockey [1,2] to basketball [3-6]. There is a considerable amount of literature in the field of baseball employing machine learning techniques [7-10]. Besides, machine learning methods have begun to examine the issue of football from various aspects [11-16].

Since the beginning of human existence, people have classified something. When people classify things, they arranged and defined them based on some parameters that they have in general. In the same way, the machines classify the data according to their characteristics. Also, it allows researchers to understand certain qualities and differences in the subject area of interest. Likewise, football clubs have certain characteristics. It is among the aims of football authorities to determine the characteristics of successful clubs and to set goals in this direction.

Empirical studies with the appropriate tools to guide soccer teams are needed to identify the characteristics of successful clubs. In this way, clubs can set new targets according to the performance criteria that constitute key distinctions through decision trees. The main reason for this study is therefore established. The other contribution of this study that there are no previous works done in this context. Accordingly, there might be new applications to extends the current literature further. Taken together, this study assesses the performance of decision tree techniques to classify European football teams in Serie A, La Liga, and Premier League—which is the first extensive examination that provides new insights into literature. We divided football teams into three groups: a) the top tiers of the leagues which finished the season pretty well to qualify European Championship League and Euro League, b) the

teams which are below top-tiers and above-average denoted as top-half, c) Teams that finish the leagues in 11th place and below. The reasoning for this type of categorization is that we perceive several commonalities in football teams in preliminary review which is also supported by Table 1. This paper uses 10-year league data consist of 600 observations. The remaining part of the work proceeds as follows. Section 2 will give information both about methodologies and the sources of data. The third section presents the preliminary statistics regarding each class of teams and the performance of each decision tree algorithms. The final section provides a summary and recommendations for further research.

2. METHOD AND DATA

2.1. Decision Trees

Classification procedure of data by machine learning divided mainly into the two-stage process. [17]. At the primary stage, the learning process constructs a model from the received knowledge. If the employer provides classification information in the learning stage, it is called supervised learning; contrarily, it is called unsupervised learning—in which classes are derived from a dataset without prior classification information [18]. A distinct advantage of using decision trees (DT) is that provides the availability to the employer to conduct both supervised and unsupervised learning. Therefore, they are commonly used for knowledge discovery [19].

The process of discovering general rules starts with DT from tuples contain classes [20]. Figure 1 displays an illustration of DT. It is clear from the top of Figure 1 that the root is a unique node— has no precursor. The remaining nodes in the DT possess precisely one precursor. There are two forms of nodes to identify: the first one is the leaves, in other sayings, the terminal nodes have no descendants; the second one is internal nodes that have more than one descendant. The test results must be equivalent to the total quantity of branches emanate from that node due to any test result associated with a single branch [20]. Lastly, the leaves contain results (i.e. classes, numbers).

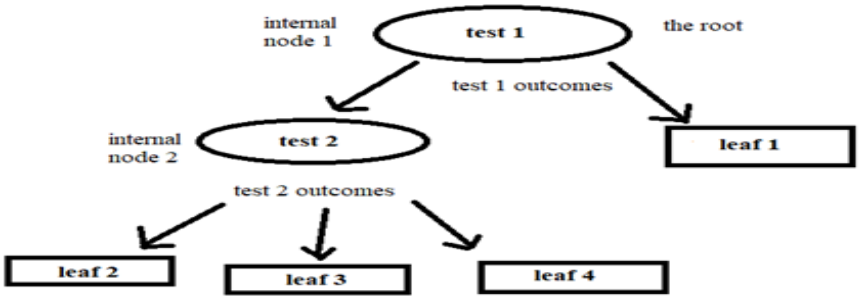


Figure 1. An Illustration of Decision Trees. Adapted from "Evolutionary Decision Trees in Large-Scale Data Mining," by [20].

Decision trees obtained through the R Software. We use the RWeka package to implement the C4.5 algorithm and Logistic Model Trees (LMT), which respectively use gain ratio and logistic regressions as division rules [21]. [22] provides detailed background information regarding the C4.5 algorithm; whereas [23] demonstrates the essentials of LMT. Cart and rpart packages are used to obtain results for the CART algorithm [24]. This simple but effective method is based on binary division by using the Gini index while creating decision rules. The random forests (RF) are another classification algorithm used in this study. Unlike the previously mentioned algorithms, the inference of overall error is proportional to the strength of each tree. It works by bringing together the predictions produced by bootstrapping a large number of independent decision trees [25]. Our reference guide for RF application is the randomForest package that [26] has introduced into the literature. The last empirical approach is regression trees based on binary division, which is similar to the Classification and Regression Trees (CART) algorithm. The classification process is based on the squared error difference between real and estimated values for the assigned dependent variable [27]. In this context, the tree library created by [28] was used for the regression tree classification process—which is mainly based on the squared error differences between real and estimated values for the assigned dependent variable.

2.2. Dataset

The data set used in the study consists of observations from Serie A, Premier League, and La Liga covering the 2009-2010 and 2018-2019 seasons obtained from the whoscored.com. There are 200 observations from each league. The separation of the data as a training dataset and test dataset was created randomly. The training data set represents 70 percent of the data. The test dataset constitutes 30 percent of the data. The classifications in the test data set are estimated by the rules learned from the training data.

Our dependent variable is denoted as "class". Teams under consideration are labeled as "top-half" in case that they completed the league in the top 10 but not well enough to participate in European Cups. The team under consideration is labeled as "top-half", in case that they finished the league in the top 10 but not well enough to participate in European Cups. The team that finished the league lower than 10th place is classified as "below-average". The number of independent variables is four. The first one is gperg which indicates the goal scored per game. The second variable is denoted as pass, reflects the successful pass percentage of the teams. The variable denoted as 'poss' shows the average ball possession percentage of football teams. Eventually, the gapg presents the number of goals conceded per game.

3. EMPIRICAL RESULTS

Table 1 below illustrates the mean value of subjected variables for each class of teams.

What stands out in the table is that the qualified teams have scored 1.86 goals per game; while top-half and below-average teams scored 1.33 and 1.07 goals respectively. The possession rate of football teams is provided in the second column, which shows qualified teams' highest mean with 54 percent. The football teams from other classes have possession of less than 50 percent. The qualified teams also

have the highest pass accuracy, which is above 80 percent, whereas the remaining teams have success below them. One notable difference between below-average football teams with the remaining classes is that they conceded a 1.57 goal per game.

Table 1. Preliminary statistics for each class of teams

Class	gperg	poss	pass	gapg
qualified	1.87	0.54	0.82	1.01
top-half	1.33	0.50	0.78	1.30
bel_ave	1.07	0.47	0.75	1.57

Turning now to the performance of each DT algorithms, Table 2 provides the accuracy rates which compares the predictions with the actual values. The calculated 95 percent confidence interval range and p-values respectively provided in the second and third row. In the case of overall accuracy rates, the random forest (RF) has the highest success in general with 79 percent. The C4.5 and CART algorithms, which have the lowest success, have 77 percent. However, if we compare the accuracy rates for qualified football teams, LMT has the highest percentage of accuracy. Unfortunately, there are disturbing results in the diagnosis of the top-half teams. The accuracy rate of All DT algorithms is below 70 percent. The most likely cause of the low accuracy rate in the top-half football teams is the need for additional data to separate them from the other classes.

Table 2. Accuracy of Decision Tree Algorithms

	C4.5	CART	LMT	Ran_Forest	Reg_Trees
Accuracy	0.77	0.77	0.78	0.79	0.78
95% CI	(0.70,0.83)	(0.70,0.83)	(0.72-0.84)	(0.72, 0.85)	(0.72,0.84)
P-Value	1.529e-13	1.529e-13	3.551e-15	9.495e-16	3.551e-15
Acc. qualified	0.84	0.86	0.88	0.85	0.86
Acc. top-half	0.59	0.62	0.65	0.68	0.63
Acc. bel_ave	0.87	0.85	0.85	0.87	0.87

4. CONCLUSION

The classification performances of the C4.5, CART, LMT, RF, and Regression Trees algorithms were evaluated for European football teams. The data set randomly separated into two subgroups. 70 percent of the data were employed to training algorithms while the remaining 30 percent used for testing. We used accuracy as criteria for the performance evaluation of the algorithms. The results of this study have shown that decision trees have a good performance in classifying football clubs. The performance of RF is the most successful based on the accuracy criterion, the rest of the DT algorithms

have also achieved more than 77 percent accuracy. The football officials can detect and progress towards the key branching factors aroused from DTs — which differentiate qualifying football teams from the rest. Nonetheless, due to the page limitations, we are unable to include all the visuals of each decision tree. We can share datasets and codes for those who are further interested.

Notwithstanding the strong accuracy rates of DT, several issues remain. It is recommended that further research be undertaken in the following ways. A further study with more focus on other machine learning techniques, such as locally weighted naïve Bayes or OneR, would be recommended. Besides, the researchers who want can obtain different results by modifying the algorithms used in this study or by changing the selection procedure of the training data. For instance, for the CART algorithm, we determined the smallest divisional value as 4. Researchers can change this value or use 80 percent of the current data set for training and 20 percent for testing. Moreover, it would be interesting if the DT techniques should also be evaluated whether there will be effective tools for football betting.

Conclusively, it is essential to participate in tournaments organized by the Union of European Football Associations (UEFA) for all football teams in the European continent. Football clubs that participate in the European Cups prosper financially by earning millions of euros in income. Therefore, football attracts the attention of many disciplines from sociology to economics due to the high impact it generated. Empirical studies made due to the teams' desire to win increase their importance. Football clubs that do not close themselves to innovations and progress in the light of science will be successful. Concerning the consequences of the machine learning techniques for football fans is that there might be a convergence of quality between football clubs that will draw more audience.

REFERENCES

[1] Mulholland J, Jensen ST. Predicting the draft and career success of tight ends in the National Football League.Journal of Quantitative Analysis in Sports. 2014;10(4):381–96. https://doi.org/10.1515/jqas-2013-0134.

[2] Joash Fernandes, C., Yakubov, R., Li, Y., Prasad, A. K., & Chan, T. C. (2019). Predicting plays in the National Football League. Journal of Sports Analytics, 1-9.

[3] Lorenzo Calvo J, Menéndez García A, Navandar A. Analysis of mismatch after ball screens in Spanish professional basketball. International Journal of Performance Analysis in Sport. 2017;17(4):555–62. https://doi.org/10.1080/24748668.2017.1367999.

[4] Leicht AS, Gomez MA, Woods CT. Team performance indicators explain outcome during women's basketball matches at the Olympic Games. Sports. 2017 Dec;5(4):96. https://doi.org/10.3390/sports5040096 PMID:29910456

[5] Cene E, Parim C, Özkan B. Comparing the performance of basketball players with decision trees and TOPSIS. Data Science and Applications. 2018;1(1):21–8.

[6] Horvat T, Havaš L, Srpak D. The Impact of Selecting a Validation Method in Machine Learning on Predicting Basketball Game Outcomes. Symmetry. 2020;12(3):431.

[7] Freiman MH. Using random forests and simulated annealing to predict probabilities of election to the Baseball Hall of Fame. Journal of Quantitative Analysis in Sports. 2010;6(2). https://doi.org/10.2202/1559-0410.1245.

[8] Oh Y, Kim H, Yun J, Lee JS. Using Data Mining Techniques to Predict Win-Loss in Korean Professional Baseball Games. Journal of Korean Institute of Industrial Engineers. 2014;40(1):8–17. https://doi.org/10.7232/JKIIE.2014.40.1.008.

[9] Tolbert B, Trafalis T. Predicting major league baseball championship winners through data mining. Athens Journal of Sports. 2016;3(4):239–52. https://doi.org/10.30958/ajspo.3.4.1

[10] Koseler K, Stephan M. Machine learning applications in baseball: A systematic literature review.Applied Artificial Intelligence. 2017;31(9-10):745–63. https://doi.org/10.1080/08839514.2018.1442991

[11] Myers BR. A proposed decision rule for the timing of soccer substitutions. Journal of Quantitative Analysis in Sports. 2012;8(1). https://doi.org/10.1515/1559-0410.1349.

[12] Folgado H, Duarte R, Marques P, Sampaio J. The effects of congested fixtures period on tactical and physical performance in elite football. Journal of sports sciences. 2015;33(12):1238–47.

[13] Schauberger G, Groll A. Predicting matches in international football tournaments with random forests. Statistical Modelling. 2018;18(5-6):460–82. https://doi.org/10.1177/1471082X18799934.

[14] Young CM, Luo W, Gastin P, Tran J, Dwyer DB. The relationship between match performance indicators and outcome in Australian Football. Journal of Science and Medicine Sport. 2019;22(4):467-71 https://doi.org/10.1016/j.jsams.2018.09.235 PMID:30352743

[15] Baboota R, Kaur H. Predictive analysis and modelling football results using machine learning approach for English Premier League. International Journal of Forecasting. 2019;35(2):741-55.

[16] Min DK. Contribution analysis of scoring in the soccer game: using decision tree. The Korean Data & Information Science Society. 2019;30(6):1385–97. https://doi.org/10.7465/jkdi.2019.30.6.1385

[17] Han J, Pei J, Kamber M. Data mining: concepts and techniques. Elsevier; 2011

[18] Michie D, Spiegelhalter DJ, Taylor CC. Machine learning. Neural and Statistical Classification. 1994;13:1–298.

[19] Rokach L, Maimon OZ. Data mining with decision trees: theory and applications. World scientific; 2008.

[20] Kretowski M. Evolutionary Decision Trees in Large-Scale Data Mining. Springer International Publishing; 2019. https://doi.org/10.1007/978-3-030-21851-5

[21] Hornik K, Buchta C, Zeileis A. Open-source machine learning: R meets Weka. Computational Statistics. 2009;24(2):225–32.

[22] Quinlan JR. Improved use of continuous attributes in C4. 5. Journal of artificial intelligence research.1996;4:77–90. https://doi.org/10.1613/jair.279

[23] Ripley B. (2019). tree: Classification and Regression Trees. R package version 1.0-40. https://CRAN.R-project.org/package=tree

[24] Therneau T, Atkinson B, Ripley B. (2019). rpart: Recursive Partitioning and Regression Trees. R package version 4.1-15. https://CRAN.R-project.org/package=rpart

[25] Breiman L. Random forests. Machine learning. 2001;45(1):5-32.

[26] Liaw A, Wiener M. Classification and regression by randomForest. R News. 2002;2(3):18–22.

[27] Loh WY. Classification and regression trees. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. 2011;1(1):14–23. https://doi.org/10.1002/widm.8.

[28] Ripley B. (2019). tree: Classification and Regression Trees. R package version 1.0-40. https://CRAN.R-project.org/package=tree

Journal of Soft Computing and Artificial Intelligence
JSCAI, 1(2): pp:92-99 (2020), www.dergipark.org.tr/en/pub/jscai
JSCAI
ISSN: 2717-8226
Received: November 27 2020 Accepted: December 19 2020

Categorization of Qualifying Football Clubs for European Cups with Backpropagating Artificial Neural Networks

[1]Bünyamin Fuat Yıldız

[1]M.Sc. (Econ.), 24 Crooks Ave Apt 229, Clifton, NJ 7011-1614 United States

Corresponding author: Bünyamin Fuat Yıldız e-mail: bunyaminfuatyildiz@yahoo.com

ABSTRACT European cups are the most popular and most profitable football organization in the world. The participation of football clubs in the Champions League and the Europa League is, therefore, a matter of interest to all parts of society. In this respect, this paper uses backpropagating ANNs to understand the capability of categorizing football clubs from Italy, England, and Spain. The sample consists of 10 years of data from Seri A, English Premier League, and La Liga — and teams categorized as qualified and unqualified. As a result of the test, backpropagating ANN classifies the clubs with 92.7 percent accuracy. Our model correctly categorized 40 of 51 qualified teams in our test dataset—that is approximately 78 percent accuracy. However, our backpropagating ANN provides more significant accuracy while predicting unqualified teams, that is approximately 98.5 percent. The probable reason for lower accuracy in the categorization of qualified teams might be underrepresentation in the dataset and lack of variable diversity. The success of ANNs implies that it could be interesting to integrate ANNs into an online betting platform to develop solutions for more complex events by introducing more data. The application of other machine learning approaches will contribute to the literature and provide an opportunity to compare methods.

KEYWORDS: Machine Learning, Backpropagated Artificial Neural Networks, Football

1. INTRODUCTION

Although oil was an influential source in the last hundred years, data seems to take its place. Nevertheless, it's not enough to merely have masses of data. It is essential to extract information from this meaningless bulk of data by applying different techniques. In this context, one of the most effective methods to turn data into information based on machine learning. Especially, artificial neural network algorithms are skilled at extracting meaningful information and efficient predictions from cumbersome amounts of data. Therefore, it has become a workhorse of researchers from different branches that produce effective solutions from customer loss analysis to cancer detection. However, there are fewer implementations of machine learning techniques in football—which creates high economic benefit for the economies.

It is the goal of football clubs to earn millions of euros in broadcasting and sponsorship revenues, as well as prestige, by participating in European Cups. Therefore, high commercialization has made football no longer an ordinary sport and makes it the focus of scientific research. Besides, various empirical studies have been carried out to achieve optimal performance. Nonetheless, machine learning applications do not have a very long history in football. One well-known early work that is often cited in research used Bayesian nets to investigate the outcome of the match result of a single team [1]. There is a great deal of recent football literature that focuses on tactical knowledge [2-4]. The number of studies on betting in football, which will have widespread popularity in the future, are relatively few

and mostly based on offline data [5-8]. [9] conducted a big data analysis to detect hooliganism in stadiums which will probably reduce the vandalism in stadiums. Studies involve various approaches for evaluating footballers' values on the transfer market [10-11]. [12] categorized European football teams into three sub-groups and investigated the categorization power of decision trees to understand certain qualities of each classified club.

In the context of the work of [12], it is possible to classify football clubs depending on their specific characteristics. If the classification process is carried out successfully, common characteristics of football clubs can be revealed. This study was conducted to evaluate the categorization performance of backpropagation ANNs —which aimed to infer common characteristics of football clubs for the qualifying teams in European Cups. The efficiency of artificial neural networks (ANNs) was tested for the first time to predict whether football clubs would be in the category to participate in European cups. In this respect, the classification made in the output layer into two groups: a) the qualified football clubs which finished the season well to qualify European cups b) the remaining football clubs denoted as unqualified. The dataset consists of 200 observations from each league. The number of ANN input layer nodes: goal per game (gperg), conceded goals per game (gapg), successful pass percentage (pass), and game possession (poss). Apart from being the first in the literature, this study might initiate new studies on the categorization of football clubs in this field. The following parts of this paper presented as follows. Section 2 contains information regarding the method and the data. In the first half of section 3, preliminary statistics shared, and then the result of the application of artificial neural networks was evaluated. The conclusion of the study is provided in the end.

2. METHOD AND DATA

2.1. Artificial Neural Networks

The technique of ANNs was developed by inspiring the relationship between the neuron and nervous system in the human brain and became popular in various fields of studies[13-14]. Information acquisition is achieved by creating various configurations with several mathematically structured neurons. By combining mathematical neurons with a variety of input vectors, a "neural network" could be configured. In all of the studies that have been carried out so far, neurons form parallel layers. Although the neuron structure has the same type of transfer function, each neuron has a weight matrix and a bias vector with a diversified set of values corresponding to the input fit of a threshold value [15]. Given the equivalent arrangement of the neurons and neural layer, the essential knowledge process and update the weights and biases. Typically, each weight matrix and vector of biases commences with arbitrary values, so as not to generate any bias in the neural network. In this situation, learning rules are necessary to get rid of randomness systematically and achieve meaningful patterns. This need is met by the learning rules that emerge as a result of training the network with data that updates weights and biases [16]. Thus, the smallest structure that makes calculations to classify the properties of the inputs called perceptron is formed [17].

The method utilized in this research is the backpropagating ANN algorithm. In the current literature, backpropagation algorithms have remained popular because of their ease of application and their multi-functionality. The main benefit of using this algorithm is to propagate the error between the output value and the actual output value obtained during the training of the network back along with their weights. When the output produced falls below a certain error value, the update process is terminated by assuming that the algorithm has learned. The special issue in the working principle of the backpropagating algorithm is that the network evaluates the output it produces with real outputs and updates each iteration until it reaches the desired error rate level. To implement the backpropagating ANNS, the neuralnet package was used which is developed by [18] in the R programming language. Since the number of hidden layers not determined by precise rules, The number of layers with the highest predictive performance will be reported. (Basheer & Hajmeer, 2000). Application of backpropagating ANN algorithm with the processor 2.60GHz i5-4300 CPU with 8192MB RAM. Accordingly, it is possible to obtain various performances with a different number of layers and nodes on computers with different features. To summarize the process of the ANN algorithm structure, inferences made using functional relationships in three stages: (i) it takes the inputs, (ii) sends it to the hidden layer (iii) and generates outputs. Besides, 70 percent of the data set was randomly allocated for training, and the remaining part for performance testing.

2.2. Dataset

The dataset was gathered from statistics sections of whoscored.com covering 600 observations from Premier Leauge, Serie A, and La Liga between the seasons 2009-2010 and 2018-2019. The dependent variable is denoted as "class". The football clubs playing in the mentioned leagues are divided into two categories. The clubs that have finished the league well enough to qualify for European Trophies classified as qualified. The rest of the clubs were categorized as unqualified. Four different independent variables were used in the learning process. These are the goals per game (gperg), the average percentage of successful passes per game (pass), game dominance (poss), and an average conceded goal per game (gapg).

3. EMPIRICAL RESULTS

3.1. Preliminary Statistical Information

In this sub-section, all analyses and graphs were carried out using Stata software. Descriptive statistics for all variables were presented in Table 1. Table 1 includes the number of observations (obs) in the second column. It is clear from the number of observations that there are no missing values. To show mean, minimum (min.), Maximum (max.), and standard deviation (SD). To reflect the properties of the data set, means, standard deviation (SD); and minimum (min) and maximum (max) values were shared.

Goal per game has a mean of 1.35 with 0.45 standard deviation. Minimum and maximum values are 0.57 and 3.18. The possession has a mean of 0.49 by 0.04 standard deviation ranging from 0.39 to 0.67. The successful pass percentage ranges from 0.62 to 0.89, with a mean of 0.77 with a 0.05 standard deviation. The number of average goals conceded per match has a mean 1.35 minimum 0.47 maximum of 2.47 and 0.33 standard deviations on average. Moreover, there are no extremities in the data set. So, there is no need for additional operation for normalization.

Table 1. Preliminary statistics for each class of teams

Variables	Obs	Mean	SD	Min	Max
gperg	600	1.35	0.45	0.57	3.18
poss	600	0.49	0.04	0.39	0.67
pass	600	0.77	0.05	0.62	0.89
gapg	600	1.35	0.33	0.47	2.47

Table 2 provides pairwise correlation coefficients for subjected variables to assess the strength of the relationship. The first column is aimed to demonstrate correlations among goals per game with other variables. As shown in Table 2, the goal per game is most highly correlated with average ball possession per game. The correlation coefficient of a goal per game with the successful pass percentage is 0.64. However, goal per game has only a negative correlation with conceded goals per game. Interestingly, the correlation between ball possession and a successful pass percentage has the strongest correlation. The negative relationship between possession percentage and conceded goals is -0.58 which is the second-highest negative relationship presented on the table. Lastly, the correlation coefficient between conceded goals and successful pass percentage is -0.51.

Table 2. Pairwise Correlations

Variables	gperg	poss	pass	gapg
gperg	1.00	—	—	—
poss	0.75	1.00	—	—
pass	0.64	0.83	1.00	—
gapg	-0.67	-0.58	-0.51	1.00

Since the most important measure of success in football is to score, the figure below is also shared. In Figure 1, a graph of the quadratic ordinary least squares regression estimates was shared to reflect the relationship of the goal scored per game with other variables. In the regression determined quadratically between the variables, the outlier numbers outside the confidence interval were found to be relatively few. When we consider the left pane of figure 1, the average number of goals scored per game has a negative linear relationship with the average number of goals conceded per game. Also, in

the middle of Figure 1, there is a positive linear relationship between the percentage of successful passes and goals scored per game. On the right side of Figure 1, possession appears to be a significant factor in average goals scored per game.

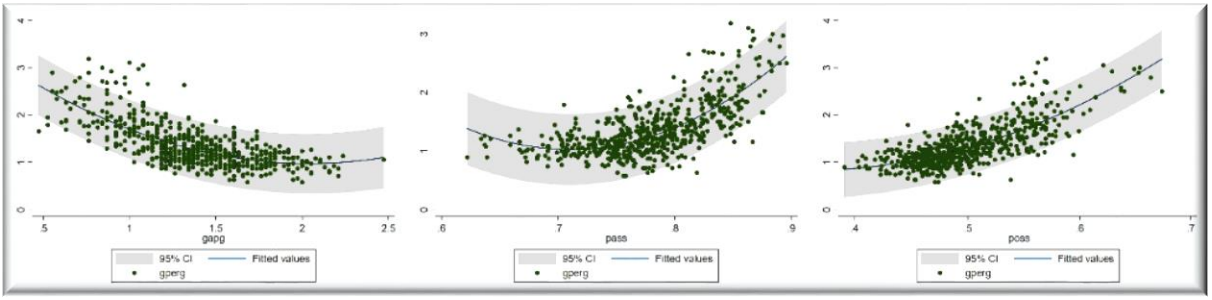


Figure 1 *The quadratic OLS plot between goal per game and other variables*

3.1. Result of Artificial Neural Networks Application

Figure 2 displays the result of ANN giving the optimum output. As seen in Figure 2, two hidden layers are consisting of 3 and 1 neurons respectively. The weight is calculated per synapse showing the influence of the related neuron, and the relevant data is sent as signals to the neural network. Since the second hidden layer with the single neuron helps in the performance of the algorithm two hidden layers were used.

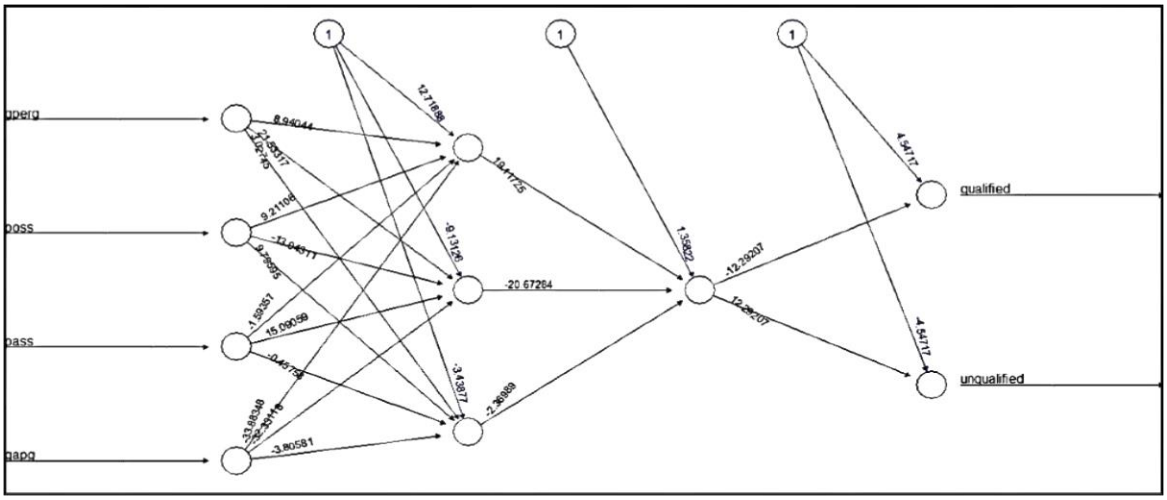


Figure 2 *The ANN plots.*

Table 3 gives the results of the confusion matrix to show the classification performance. Accuracy is the rate of correctly categorized cases compared with the number of whole cases. The accuracy of the ANN application is 92.7 percent. Out of the 51 qualified teams in the test data set, 40 were correctly predicted; however, 11 of the teams were unqualified. This constitutes a success percentage of about 78 percent. Our ANN model has achieved very successful results in the categorization of unsuccessful teams. Out of the 128 unsuccessful teams, only 2 were wrongly predicted. This corresponds to an accuracy rate of about 98.5 percent. The No Information Rate (NIR) reflects the accuracy achieved

when the prediction option is used in the direction of the majority-forming category. The NIR is 0.71, as unqualified teams make up 71 percent of the data. The "95% CI" indicates the range of values of the accuracy level within the 95 percent confidence interval. The smallness of the value obtained in P-Val [Acc> NIR] makes it easier to decide on the statistical reliability of the applied method. The p-value is extremely small shows that there is certainly no possibility that the accuracy of NIR (71%) is higher than the accuracy of the ANN (92%).

Table 3. Confusion Matrix and Statistics

CONFUSION MATRIX		**Reference Values**	
		qualified	unqualified
Predicted Values	qualified	40	2
	unqualified	11	126
Accuracy	0.927	**NIR**	0.71
95% CI	0.87, 0.96	**P-Val [Acc>NIR]**	1.337e-12

4. CONCLUSION

This study set out to evaluate how effective the backpropagating ANN algorithm classifies football clubs in the three major European leagues. The application aims to determine the status of teams in the English Premier League, Seri A, and La Liga whether to participate in European Cups. Seventy percent of the available data were selected to train the ANN. The accuracy of the model was checked with the test data, following the training process. The results of the test show that the accuracy of the ANN was 92.7%. Note that while the accuracy rate is lower for qualified teams, the prediction success of unqualified teams is higher. There could be two probable reasons for this. The success of clubs constitutes a more complex form. Therefore, it might difficult to categorize qualified teams with the existing variable set. The second possible reason may be due to the underrepresentation of qualified European football clubs in our data set.

These results contribute to the rapidly expanding field of machine learning applications in the literature of football. However, this research has thrown up many topics in need of further investigation. The methods used for this analysis may be applied to other leagues elsewhere in the world. Besides, it would be interesting to repeat the experiments described here with different techniques such as support vector machines. Hence, it is possible to highlight the difference between the accuracy of techniques. All in all, techniques like artificial neural networks are used effectively in all disciplines; they are relatively new in football. The limitations of this study predictions are made from historical information. Determining the result by live performance indicators might probably be more interesting. Also, the performance of the teams at the end of the league could be predicted from data such as player performances at the start of the league.

Concerning the implications of the new technological tools for football fans, closing the gap between football clubs can be achieved as a consequence of coworking sports scientists and data analysts.

REFERENCES

[1] Joseph A, Fenton NE, Neil M. Predicting football results using Bayesian nets and other machine learning techniques. Knowledge-Based Systems. 2006;19(7):544-53.

[2] Rein R, Memmert D. Big data and tactical analysis in elite soccer: future challenges and opportunities for sports science. SpringerPlus. 2016 Dec;5(1):1-3.

[3] Berrar D, Lopes P, Dubitzky W. Incorporating domain knowledge in machine learning for soccer outcome prediction. Machine learning. 2019 Jan 15;108(1):97-126. https://doi.org/10.1007/s10994-018-5747-8.

[4] Herold M, Goes F, Nopp S, Bauer P, Thompson C, Meyer T. Machine learning in men's professional football: Current applications and future directions for improving attacking play. International Journal of Sports Science & Coaching. 2019 Dec;14(6):798-817. https://doi.org/10.1177/1747954119879350.

[5] Bunker RP, Thabtah F. A machine learning framework for sport result prediction. Applied computing and informatics. 2019 Jan 1;15(1):27-33.

[6] Hubáček O, Šourek G, Železný F. Learning to predict soccer results from relational data with gradient boosted trees. Machine Learning. 2019;108(1):29-47. https://doi.org/10.1007/s10994-018-5704-6.

[7] Stübinger J, Mangold B, Knoll J. Machine Learning in Football Betting: Prediction of Match Results Based on Player Characteristics. Applied Sciences. 2020;10(1):46. https://doi.org/10.3390/app10010046.

[8] Rudrapal D, Boro S, Srivastava J, Singh S. A Deep Learning Approach to Predict Football Match Result. InComputational Intelligence in Data Mining 2020 (pp. 93-99). Springer, Singapore.

[9] Fenil E, Manogaran G, Vivekananda GN, Thanjaivadivel T, Jeeva S, Ahilan A. Real time violence detection framework for football stadium comprising of big data analysis and deep learning through bidirectional LSTM. Computer Networks. 2019;151:191-200. https://doi.org/10.1016/j.comnet.2019.01.028.

[10] Matesanz, D., Holzmayer, F., Torgler, B., Schmidt, S. L., & Ortega, G. J. (2018). Transfer market activities and sportive performance in European first football leagues: A dynamic network approach. PloS one, 13(12), e0209362.

[11] Singh P, Lamba PS. Influence of crowdsourcing, popularity and previous year statistics in market value estimation of football players. Journal of Discrete Mathematical Sciences and Cryptography. 2019 17;22(2):113-26. https://doi.org/10.1080/09720529.2019.1576333.

[12] Yıldız BF. Applying Decision Tree Techniques to Classify European Football Teams. Journal of Soft Computing and Artificial Intelligence. 2020; 1(2): 29-35.

[13] Kaplan K, Kuncan M, Ertunc HM. Prediction of bearing fault size by using model of adaptive neuro-fuzzy inference system. In2015 23nd Signal Processing and Communications Applications Conference (SIU) 2015 May 16 (pp. 1925-1928). IEEE.

[14] Bayram, S., Kaplan, K., Kuncan, M., & Ertunç, H. M. (2014, April). The effect of bearings faults to coefficients obtaned by using wavelet transform. In *2014 22nd Signal Processing and Communications Applications Conference (SIU)* (pp. 991-994). IEEE.

[15] Cavuto DJ. An exploration and development of current artificial neural network theory and applications with emphasis on artificial life. Unpublished Master of Engineering Thesis. The Cooper Union, Albert Nerken School of Engineering, New York, NY. 1997.

[16] Dreyfus SE. Artificial neural networks, back propagation, and the Kelley-Bryson gradient procedure. Journal of guidance, control, and dynamics. 1990;13(5):926-8. https://doi.org/10.2514/3.25422.

[17] Basheer IA, Hajmeer M. Artificial neural networks: fundamentals, computing, design, and application. Journal of microbiological methods. 2000 1;43(1):3-1.

[18] Günther F, Fritsch S. neuralnet: Training of neural networks. The R journal. 2010 1;2(1):30-8.