CAIAC

INTERNATIONAL CUMHURIYET ARTIFICIAL INTELLIGENCE APPLICATIONS CONFERENCE 2021

3rd-4th DECEMBER | ONLINE

http://caiac.cumhuriyet.edu.tr

CONFERENCE PROCEEDINGS

SİVAS CUMHURİYET ÜNİVERSİTESİ 1974

SİVAS CUMHURİYET ÜNİVERSİTESİ YAPAY ZEKA VE VERİ BİLİMİ UYGULAMA VE ARAŞTIRMA MERKEZİ

aselsan

CONFERENCE PROCEEDINGS

ISBN: 978-605-7902-60-3

CAIAC'21

The Proceedings of The International Cumhuriyet Artificial Intelligence Applications Conference

2021

Edited By

Assoc. Prof. Ahmet Gürkan YÜKSEK

Prof. Serkan AKKOYUN

Dr. Emre DELİBAŞ

Dr. Abdulkadir ŞEKER

Saliha YEŞİLYURT

Published

Sivas Cumhuriyet University, February 2022,

caiac@cumhuriyet.edu.tr

WELCOME TO CAIAC'2021

International Cumhuriyet Artificial Intelligence Applications Conference'2021 (CAIAC'2021) will provide an excellent international forum for sharing knowledge and results in theory, methodology and applications of Artificial Intelligence. The Conference looks for significant contributions to all major fields of the Artificial Intelligence, Soft Computing in theoretical and practical aspects. The aim of the Conference is to provide a platform to the researchers and practitioners from both academia as well as industry to meet and share cutting-edge development in the field.

Authors are solicited to contribute to the Conference by submitting articles that illustrate research results, projects, surveying works and industrial experiences that describe significant advances in the listed areas but are not limited to.

CAIAC is organized with the cooperation of Sivas Cumhuriyet University Artificial Intelligence and Data Science Applications and Research Center and ASELSAN Research Center.

Assoc. Prof. Ahmet Gürkan YÜKSEK
Chairman of Conference

ORGANIZATION COMMITTEE

Dr. Ahmet Gürkan YÜKSEK	Sivas Cumhuriyet University
Dr. Serkan AKKOYUN	Sivas Cumhuriyet University
Dr. Emre DELİBAŞ	Sivas Cumhuriyet University
Dr. Abdulkadir ŞEKER	Sivas Cumhuriyet University
Saliha YEŞİLYURT	Sivas Cumhuriyet University
İbrahim BÜYÜKGENÇ	ASELSAN
Cemran ÜNAL	ASELSAN
Avni Selim ÖZÇUKURLU	ASELSAN

SCIENTIFIC COMMITTEE

Dr. Alpaslan Fığlalı	Kocaeli University)
Dr. Andrew Kusiak	The University of Iowa
Dr. Bahriye Akay	Erciyes University
Dr. Banu Diri	Yıldız Technical University
Dr. Çetin Elmas	Azerbaijan Technical University
Dr. Ecir Uğur Küçüksille	Süleyman Demirel University
Dr. Ferhat Sayım	Yalova University
Dr. Haldun Akpınar	Marmara University
Dr. Halife Kodaz	Konya Technical University
Dr. Harun Uğuz	Konya Technical University
Dr. Manafəddin Namazov	Baku Engineering University
Dr. Mehmet Ali Alan	Sivas Cumhuriyet University
Dr. Mehmet Göktürk	Gebze Technical University
Dr. Oğuz Kaynar	Sivas Cumhuriyet University
Dr. Şadi Evren Şeker	Antalya Bilim University
Dr. Vasıf Nabiyev	Karadeniz Technical University
Dr. Yusuf Sinan Akgül	Gebze Technical University
Dr. Celal Öztürk	Erciyes University
Dr. Çiğdem Erol	Istanbul University
Dr. Emre Dandıl	Bilecik Seyh Edebali University
Dr. Ferdi Sönmez	Fenerbahçe University
Dr. Güzin Ulutaş	Karadeniz Technical University
Dr. Hidayet Takçı	Sivas Cumhuriyet University
Dr. İ. Berkan Aydilek	Harran University
Dr. İhsan Hakan Selvi	Sakarya University
Dr. Konstantin P. Katin	National Research Nuclear University MEPhI
Dr. M. Amaç Güvensan	Yıldız Technical University
Dr. Muhammed Kürşad UÇAR	Sakarya University
Dr. Selçuk Ökdem	Erciyes University
Dr. Süleyman Eken	Kocaeli University
Dr. Ahmet Özkış	Necmettin Erbakan University
Dr. Buket Karatop	İstanbul University-Cerrahpaşa
Dr. Emre Yalçın	Sivas Cumhuriyet University
Dr. Eyüp Çalık	Yalova University
Dr. Feriştah Dalkılıç	Ege University
Dr. Fırat İsmailoğlu	Sivas Cumhuriyet University
Dr. Gülgönül Bozoğlu Batı	Yalova University
Dr. Güneş Harman	Yalova University
Dr. Kali Gürkahraman	Sivas Cumhuriyet University
Dr. Metin Bilgin	Uludağ University
Dr. Metin Zontul	İstanbul Arel Üniversity

Organized by

Sponsored by

PANEL SPEAKERS

Panel Title: Artificial Intelligence in Healthcare

Prof. Bahriye AKAY / Erciyes University
"Challenges with AI-Based Healthcare Solutions"

Asst. Prof. Özkan Ufuk NALBANTOĞLU / Erciyes University
"Bioinformatics, Artificial Intelligence and Next Generation Medicine"

Panel Title: Artificial Intelligence in the Defense Industry and Future Projects

Moderator: Veysel YÜCESOY, PhD / ASELSAN, Manager of AI Research Center

SECTION-1: In context of Natural Language Processing
Prof. Yusuf Sinan AKGÜL / Gebze Technical University
Assoc Prof. Mehmet Fatih AMASYALI / Yıldız Technical University
Çağrı TORAMAN, PhD / ASELSAN
Mehmet Emre ATASOY / ASELSAN

SECTION-2: In context of Decision Support System
Prof. Murat Osman ÜNALIR / Ege University
Ümitcan ŞAHİN / ASELSAN
Miase ÖRÜMLÜ / ASELSAN

SECTION-3: In context of Computer Vision
Asst. Prof. Alper KILIÇ / Konya Technical University
Mustafa AYAZOĞLU / ASELSAN
İbrahim Batuhan AKKAYA / ASELSAN

SECTION-4: In context of Cyber Security
Prof. Dr. Burak Berk ÜSTÜNDAĞ / İstanbul Technical University
Asst. Prof. Murat İSKEFİYELİ / Sakarya University
Ali IŞIKLI / ASELSAN

KEYNOTE SPEAKERS

Asst. Prof. Şebnem ÖZDEMİR / İstinye University
"From Universe to Meta Universe: New Threats and New Opportunities with Artificial Intelligence"

Assoc. Prof. Konstantin KATİN / National Research Nuclear University MEPhI (Moscow, Russia)
"Machine Learning as a Bridge From ab Initio Calculations to Prediction of Experimentally Measured Quantities"

Prof. Vasif NABİYEV / Karadeniz Technical University
"AI From Past to Today: Problems, Solutions, Issues"

CONTENTS

Segmentation of Histological Images using Artificial Bee Colony Algorithm

Khalif Muse WARSAME
Dept. of Computer Engineering,
Erciyes University
Kayseri, Turkey
khalifmuse515@gmail.com

Rıfat KURBAN
Dept. of Computer Technologies,
Kayseri University
Kayseri, Turkey
rifatkurban@kayseri.edu.tr

Ali DURMUŞ
Dept. of Electricity and Energy,
Kayseri University
Kayseri, Turkey
alidurmus@kayseri.edu.tr

***Abstract*— Image segmentation is an important topic in image processing. Various algorithms are proposed for segmentation of medical images. In this paper, we study 3 methods for color image segmentation, particularly histological images stained with Hematoxylin and Eosin. Color image segmentation is a process of extracting information from the images more connected regions satisfying uniformity criterion which is based on features derived from spectral component. Color images provide more information than gray version of it. In this paper, images obtained from the literature are segmented by K-means, fuzzy C-means and artificial bee colony (ABC) algorithm based iterative Euclidean distance minimization. Experiments are conducted both on RGB color spaces and Lab color spaces. Experimental analyses show that methods achieve successful segmentation results.**

***Keywords*— Histological image segmentation, K-means, Fuzzy c-means, ABC (artificial bee colony).**

I. INTRODUCTION

Pathology analyzes the change in cell structure and tissue distribution by using microscope and then determines if a biopsy is malignant [1]. Image segmentation is a difficult and complex task in image processing, computer vision and pattern recognition applications [2,3]. Image segmentation is an essential technique in medical diagnosis through the analysis of the microscopic images. Image segmentation separates objects of interest from background through various methods in image processing such as thresholding where the intensity value is used to separate the regions [4]. Well known image segmentation methods can be grouped as: thresholding algorithms [5, 6], clustering techniques [7], compression-based methods [8], histogram-based methods [9], edge detection [10] and region-based techniques [11].

In image segmentation, the gray level of pixels of the object are very different from the gray level of the pixels of the background. Thresholding is an effective technique in separating the object from the background. The main aim of thresholding is to choose the gray-level threshold value for discriminating the target object in an image from the background in term of their gray-level distribution. In threshold technique, binary image is computed from converted gray level image. The information of the binary image should include the positions and shapes of the objects of interest [5]. The main problem of global thresholding techniques is that they cannot distinguish those pixels which share the same gray level but do not belong to the same group [6].

Clustering method is to divide data set at hand into groups are more like each other than elements in different clusters. Clustering is a method of grouping a set of patterns into several clusters such that similar patterns are assigned to one cluster. Each cluster has a well-defined centroid. The objective of clustering technique is to measure the similarity or to compute the distance between the respective patterns. Clustering methods can be categorized into two such as supervised and unsupervised. Supervised learning-based image segmentation algorithms separate the image based on the sample of object colors using prior knowledge. Unsupervised methods are automated methods, and they segment regions in feature space that have higher density [7].

In image segmentation techniques based on similarity criteria and region splitting and merging technique are very similar methods. The main aim of region splitting technique is to define similarity of the image. Region splitting technique checks the quadrants for the same defined criteria and divides it further into four quadrants if the test result is negative and the process continues till the criteria is satisfied [12].

Medical imaging becomes an essential part of health care, recently. There are many medical imaging methods are widely used such as X-rays, ultrasound, magnetic resonance imaging (MRI) and computed tomography (CT). When images are segmented perfectly, one can easily extract information from them, and this makes them simple and easy to analysis. That makes medical decision more reliable and increases patient safety.

Developing an automatic image segmentation algorithm, which aims at automated extraction of object boundary feature, plays an important role in understanding medical image contents for diagnosis and analysis. Medical image processing plays an important role in assisting health care providers for proper diagnosis and suitable treatment. In general, most of existing methods do not use adaptive optimization techniques and have a complex structure. RGB color model is also preferred instead of using other color spaces.

Recently, automatic image segmentation techniques for histological image segmentation become an essential for medical diagnosis and analysis to understand if cell is affected by the cancer or not. Motivation of this study is to develop an automatic image segmentation algorithm that can be used to segment histological images stained with Hematoxylin and Eosin (H&E) to produce an optimum segmented image that makes possible to analyze and detect easily for the cancer.

Aim of this study is to simplify representation of pathological images to make the image reliable, more useful and simply to analysis. In this paper, a simple segmentation technique based on Euclidean distance based iterative

clustering and Artificial Bee Colony algorithm on histological images using both RGB and Lab color space is developed.

II. COLOR SPACES AND SEGMENTATION ALGORITHMS

A. RGB and Lab Color Spaces

A color space is a systematic way that can be representation of a set of colors. Color space is a technique that can be specified and visualize existing colors. Humans can describe a color by its hue (color), saturation and brightness (luminosity). However, computers define a color by using its amount of RGB phosphor emission required to match colors. Clustering method scan used to define similarity clusters of elements in the feature space and label cluster as a different region. Color space allows for reproducible representations of color, in analog and digital representations.

The Lab space is useful for sharpening images and the removing artifacts in JPEG images or in images from digital cameras and scanners. The Commission Internationale del'Eclairage (CIE) has been developed this color space. Lab color space is a color opponent space has three components L (lightness), a (involved red and green colors) and b (controls blue and yellow colors). It has non-linear coordinates and is designed to approximate human vision. Also, it is used as an interchange format between different devices. Lab model is three dimensional and can only presented in a three-dimensional space [13].

B. Segmentation Methods

Partitioning an image into a set of regions that can cover most of the image is called as image segmentation. It is the task of partitioning an image into similar areas or groups. Changing representation and simplifying an image into something useful and easier to diagnose that is the goal of an image segmentation technique. An unsupervised method is an automatic clustering that can be grouping unleveled data set into clusters. Segmentation techniques can be classified into the following basic concepts: pixels-based, edge-based, cluster-based, region-based, color-based and hybrid methods.

III. IMAGE SEGMENTATION USING ABC ALGORITHM

In this study, a method based on ABC optimization algorithm is compared with well-known methods.

A. Artificial Bee Colony (ABC) Algorithm

Karaboga proposed an efficient optimization technique using less parameter compared with others [14]. Artificial Bee Colony (ABC) algorithm compared to other population techniques, ABC demonstrates simplicity and flexibility. In ABC, food position stands a solution to the problem and nectar represents the objective function of the solution. Artificial bees are grouped into three: employed bees, onlooker bees, and scout bees. Each employed bee carries with her information about the associated food source and shares the information to onlookers. Onlooker bees wait in the hive on the dance area for the information about the possible food sources, and then make decision to choose a food source in order to exploit it. Onlooker bees select a food source related with the quality. Food sources with high quantity of nectar attract more onlooker bees compared to ones with a lower quantity of nectar. The employed bee whose food source has been exhausted becomes a scout bee which explores a possible promising food source in the neighborhood of the hive. In ABC algorithm, the number of employed and onlooker bees are equal.

B. Proposed Segmentation Method

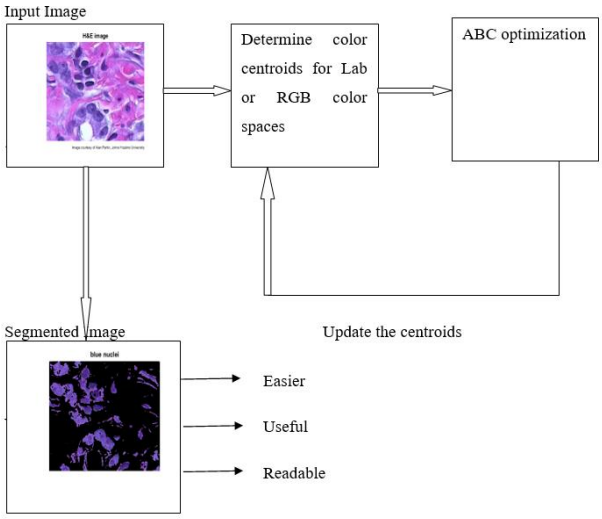


Fig. 1. Flowchart of the proposed segmentation method.

An iterative segmentation technique based on optimizing the distance of sub-neighbors of each centroid by using ABC is proposed. The method can use both RGB and Lab color values of the pixels. The block diagram of the method is given in Figure 1. The main steps of the algorithm can be summarized as follows:

1) Read RGB image and decomposed R, G and B channels.
2) Convert image from RGB color space to Lab color space and decompose L, a and b channels.
3) Choose either to use RGB or Lab color space.
4) Segment the pixel values by using ABC:
   a) Generate initial random 3 centroids because of the images are composed of 3 parts: cell nuclei, cytoplasm and the remaining. Centroids are initialized in continuous space in the range of [0,255].
   b) Calculate the Euclidean distance of each pixel color value the generated centroids: $D_j = \sum_{i=1}^{N}(P_i - C_j)^2$ (1) where N is the total count of pixels, P is a pixel's color value, C is a generated centroid and j={1,2,3}.
   c) Find the $E_i = \min(D_j)$ for each pixel and assign C_j to L_i where D_j is minimum. E is the error vector for all pixels and L is the segmented image.
   d) Minimize the total error of each pixel by updating the centroids C using employed, onlooker and scout bees.
   e) Repeat the steps until a predefined cycle number is met.
5) Use ED as the error metric and L as the segmented image.

IV. EXPERIMENTS AND RESULTS

K-means, Fuzzy C-means and the ABC based method are used to segment histological images stained with H & E, and the results are compared. Also, sum of Euclidean distance errors (ED), mean squared error (MSE) between the original image and segmented the image and similarity between original images and segmented image (SSIM) are also calculated. The test images in the dataset are summarized in Table I.

TABLE I. DESCRIPTION OF IMAGES DATASET USED IN THE EXPERIMENTS

Image Number	Description
Image #1	Size 200 x 132 Granular cell variant of clear cell renal cell carcinoma on hematoxylin and e stain. Institution Medscape [15]
Image #2	size 586 x 345 Cross-section of multi-cellular bio printed human liver tissue, stained with hematoxylin and Eosin (H&E) [16]

Parameter settings for the methods used in the experiments are given in Table II. The parameters are the default values of the methods.

TABLE II. PARAMETER SETTINGS FOR DIFFERENT CLUSTERING METHODS.

K-means		Fuzzy C-means		ABC	
Parameter	**Value**	**Parameter**	**Value**	**Parameter**	**Value**
Distance	Euclidean	Exponent for U	2	Pop. size	32
EmptyAction	Singleton	Max Iter	100	Max. cycle number	540
Max Iter Start	100 Plus	Min. Amout of Improvement	1e-5		

Experiments have been conducted on a desktop computer which have the following specifications: AMDX6 1055 six core CPU, 8GB of RAM and Windows 2012 Server operating system. Methods are implemented by using MATLAB programming language. Experiments are repeated 30 times due to the usage of stochastic methods.

In Figure 2, two test images stained with H&E are visualized. These images are used in the experiments to compare the algorithms. The images used are encoded at 24-bit true color RGB. In the experiments, both RGB and Lab channels are used for all methods.

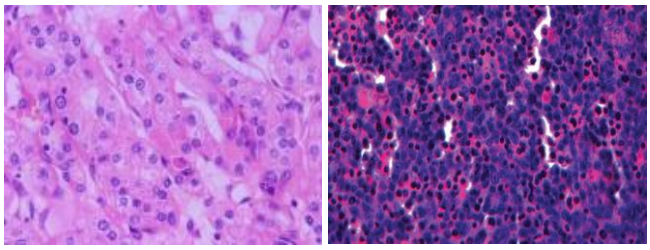


Image #1 Image #2

Fig. 2. Test images used in the experiments.

Figure 3 shows segmented images results. K-means, Fuzzy C-Means and ABC based clustering methods are used for both RGB and Lab channels. Methods clusters the images into three colors.

Table 3 illustrates the results of the two test images. The bold results show which method has better result than the other methods. In Table 3, ED, MSE and SSIM results of all algorithms are given. For ED and MSE, small values are better. For SSIM higher values are better. Quantitative results show that ABC based method is slightly better than the others. The number in the parenthesis indicates the standard deviation of 30 runs.

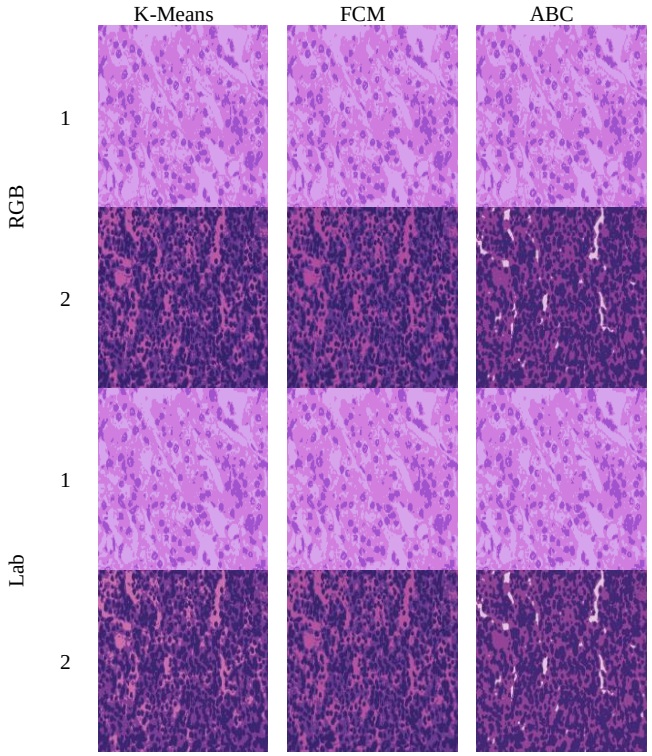


Fig. 3. Comparison of segmented images obtained with K-means, Fuzzy C-means and ABC methods

TABLE III. QUANTITATIVE RESULTS OF THE SEGMENTATION EXPERIMENTS.

			K-means	Fuzzy C-means	ABC-ED
Image #1	**RGB**	ED	141.309(0.000)	141.593(0.000)	**141.308(0.001)**
		MSE	226.850(0.071)	228.047(0.000)	**226.809(0.068)**
		SSIM	1.364(0.000)	1.361(0.000)	**1.365(0.000)**
	LAB	ED	353.887(0.000)	354.344(0.000)	**353.886(0.000)**
		MSE	226.340(0.000)	**225.953(0.000)**	226.340(0.009)
		SSIM	1.389(0.000)	1.387(0.000)	**1.390(0.001)**
Image #2	**RGB**	ED	505.498(12.684)	530.581(0.000)	**479.843(0.001)**
		MSE	941.319(16.260)	**923.256(0.000)**	974.617(0.597)
		SSIM	0.989(0.026)	**1.050(0.000)**	0.936(0.001)
	LAB	ED	1558.065(0.000)	1582.742(0.000)	**1544.946(0.040)**
		MSE	**907.836(0.006)**	927.132(0.000)	940.597(0.299)
		SSIM	1.044(0.000)	**1.065(0.000)**	0.934(0.001)

Segmentation results are visualized in the Figures 4-7. Each row of Figure 4 indicates the different parts (segments) of the images such as cell nuclei, cytoplasm or the remaining. Three segments of each image using K-means, fuzzy c-means and ABC methods for both Lab and RGB color spaces are given. In Figure 4 and 5, segmentation results of Image #1 is given for Lab and RGB color spaces, respectively.

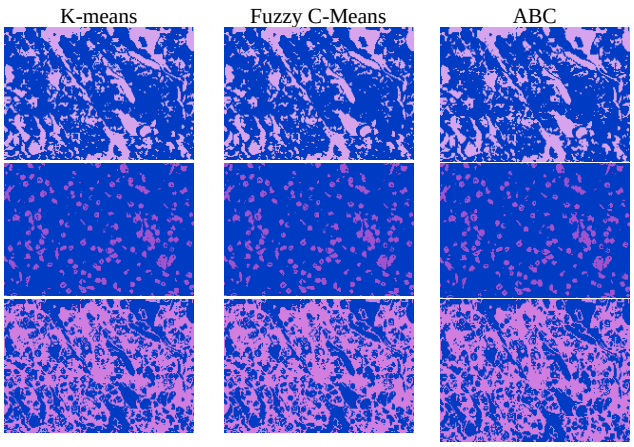


Fig.4. Visual segmentation results of Image #1 using Lab color space.

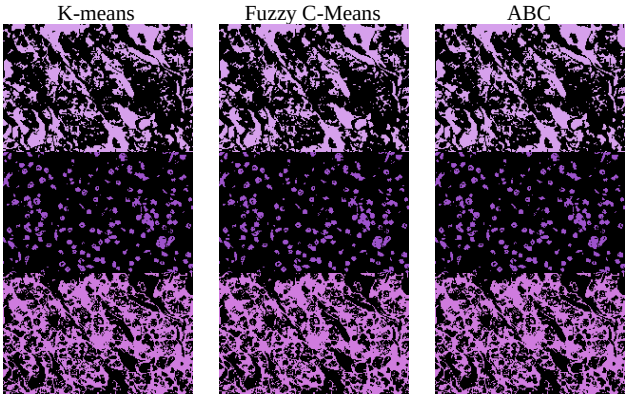


Fig.5. Visual segmentation results of Image #1 using RGB color space.

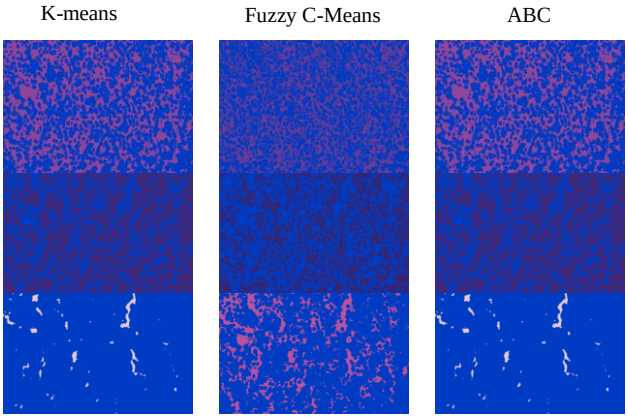


Fig.6. Visual segmentation results of Image #2 using Lab color space.

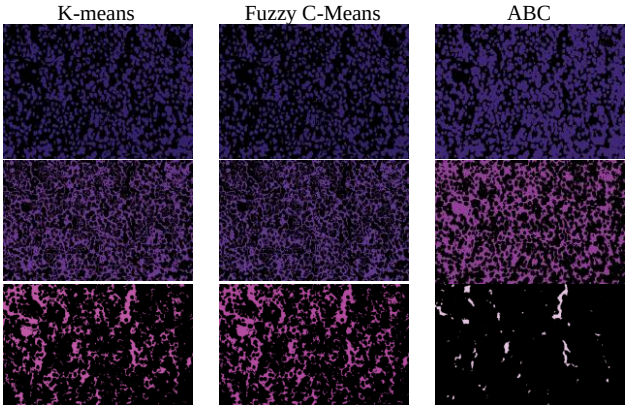


Fig.7. Visual segmentation results of Image #2 using RGB color space.

As shown in Figure 4 and 5, Lab and RGB spaces results are similar. K-means, fuzzy c-means and ABC results are similar.

In Figure 6 and 7, segmentation results of Image #2 are given for Lab and RGB color spaces, respectively.

As shown in Figure 6 and 7, Lab color space is better than RGB color space. Results of ABC method is better than K-means and Fuzzy C-Means methods.

V. CONCLUSIONS

Cancer has become major problem of the human health and the development of techniques for use in its diagnosis and treatment has increasingly become a focus of research. In this study, comparison of the three clustering techniques Fuzzy C-means, K-means and ABC optimization are compared. Fuzzy C-means and K-means are the most popular algorithms used clustering algorithms. Also, an iterative segmentation technique based on optimizing the distance of sub-neighbors of each centroid by using ABC is proposed. Image segmentation is the task of processing an image into its constituent objects. Recently, considerable work on histological segmentation has focused on fully automated approaches that rely on machine learning techniques. Color based features are important in histology applications since biologists stain tissues to highlight special structures. The color difference is used to classify specific objects to be investigated and to find relationship among them. The aim to use image segmentation technique is to find the objects of interest depending on the image and its characteristics.

ABC based segmentation aims to find nearest neighbors of each centroid to minimize the distance of each pixel to its corresponding centroid. This method is used to improve an efficient of Euclidean distance improvement by using ABC optimization algorithm. The comparative study of algorithms shows that the results are similar in most cases. However, in particular cases, ABC is slightly better than Fuzzy C-means and K-means.

Histology is the anatomical study of the microscopic structure of tissues. It is regarded as a gold standard for clinical diagnosis of diseased tissue and for the identification of therapy effects. The manual interpretation is however very time consuming and prone to subjective evaluation. The digitized tissues have become an important application of computer image analysis as well as with the use of machine learning techniques. The role of computer aided diagnosis (CAD) algorithms in medical imaging has escalated to a point where algorithms have been developed for disease detection, diagnosis, and prediction of prognosis that complements the opinion of the pathologist. Therefore, CAD systems are becoming crucial in histology analysis, as they could automatically identify regions of medical interest.

REFERENCES

[1] U. Rajyalakshmi, K. S. Prasad, and R. S. Koteswara, “Tissue processing, staining and image processing of Pathological Cancer Images A Review,” International Journal of Engineering and Advanced Technology (IJEAT), vol. 3(3), pp. 10-15, 2014.

[2] M. Xess and S. A. Agnes., “Survey on clustering based color image segmentation and novel approaches to FCM algorithm,” IJRET International Journal of Research in Engineering and Technology vol. 2(12), pp. 346-349, 2013.

[3] N. Senthilkumaran, R. Rajesh, “A study on rough set theory for medical image segmentation,”International Journal of Recent Trends in Engineering, vol. 2(2), pp. 236-238, 2009.

[4] D. Belsare, M. M. Mushrif, “Histopathological image analysis using image processing techniques an overview. Signal and Image Processing An International Journal (SIPIJ), vol. 3(4), pp. 23-36, 2012.

[5] S. S. Al-amri, N. V. Kalyankar, and ,S. D. Khamitkar, “İmage segmentation by using thershod techniques,” Journal of Computing, vol. 2(5), pp. 83-86, 2010.

[6] H. M. Som, J. M. Zain and A. J. Ghazali., “Application of threshold techniques for readibility improvement of jawi historical manuscript images.” Advanced Computing: An International Journal (ACIJ), vol. 2(2), pp. 60-69, 2011.

[7] M. C. Jobin, R. M. S. Christ Parvathi, “Magnetic resonance brain image.”, International Journal of VLSI design and Communication Systems (VLSICS), vol. 3(4), pp. 121-133, 2012.

[8] A. F. Jassim, E. H. Qassim, “Five modulus methods for image compression.” Signal and Image Processing An International Journal (SIPIJ), vol. 3(5), pp. 19-28, 2012.

[9] J. T. Orlando, and R. Seara., “Image Segmentation by Histogram Thresholding using Fuzzy Set.” IEEE Transection on Image Processing, vol. 11(12), pp. 1457-1465, 2002.

[10] N. Senthilkumaran, R. Rajesh, “Edge detection techniques for image segmentation servey of soft computing approaches” International Journal of Recent Trends in Engineering, vol. 1 (2), pp. 250-254, 2009.

[11] S. Lakshmi, V. Sankaranarayanan, “A study of edge detection techniques for segmentation computing approaches,” IJCA Special Issue on “Computer Aided Soft Computing Techniques for Imaging and Biomedical Applications” CASCT, pp. 35-41, 2010.

[12] A. M. Khan , S.Ravi, “Image Segmentation Methods: A Comparative Study,” International Journal of Soft Computing and Engineering (IJSCE), vol. 3(4), pp. 84-92, 2013.

[13] V. S. Rathore1, M. S. Kumar, A. Verma., “Colour Based Image Segmentation Using L*A*B* Colour Space Based On Genetic Algorithm.,” International Journal of Emerging Technology and Advanced Engineering, vol. 2(6), pp. 156-162, 2012

[14] D. Karaboga, B. Akay., “A comparative study of Artificial Bee Colony algorithm. Applied Mathematics and Computation,” vol. pp. 108–132, 2009.

[15] http://emedicine.medscape.com

[16] http://www.organovo.com/sites/default/files/assets/images/pic-liver2.jpg

Modeling of Non-Linear Motor with Machine Learning Algorithms

Mehmet Akif BUZPINAR
dept. Gemerek Vocational School of
Cumhuriyet University,Sivas,Turkey
makifbuz@gmail.com

***Abstract*— It is very difficult to completely define a nonlinear electric motor such as Switched Reluctance Motor (SRM) by its mathematical model. For a three-phase 6/4 SRM, current data varying according to the rotor position were obtained from the phase windings by the pulse injection method applied to the sensored driven motor. By measuring the phase currents changing according to the rotor position, the input data of the model and the active phase output data were obtained. The relationship between these current values and the rotor position was analyzed by machine learning algorithms and the rotor position was predicted. Tuned Fine Tree and Ensemble Bagged Tree algorithms were determined as the most successful algorithms in the study, in which time series approach was used to increase model success. The motor was driven at constant speed and load to keep the model size small. Data is not preprocessed to reduce microcontroller load, shorten processing time, and facilitate real-time operation. This approach eliminated the pre-mathematical processing load and contributed to the estimation speed as high as possible. In this study, it has been shown that machine learning algorithms used in modeling nonlinear systems can be used to model electric motors with similar structures and can increase model performance with time series approach.**

In nonlinear motor such as SRM, modeling motor provide sensorless driving with machine learning algorithm is a new approach. Different from the inductance and magnetic flux-based positioning approaches, the ongoing effect of the magnetic behavior has led the model training with the time series method in order to determine the position according to the instantaneous current values.

Keywords—machine learning, fine tree, ensemble bagged tree, switched reluctance motor, time series method

I. INTRODUCTION

Switched Reluctance Motor (SRM), which has an extremely non-linear behavior, is a difficult motor to mathematically model and operate with high efficiency [1]. The remarkable features of the motor can be listed as its simple and robust structure, high torque and absence of magnets. With these aspects, it can be a preferred electrical motor type for industrial applications. However, it is an acoustically and electrically noisy motor and torque fluctuation is affected by driving performance [1].

Efficient driving in SRM can be achieved with the correct modeling of the motor and correct phase control according to the rotor position. SRM, which is driven by using a mechanical rotor position sensor in sensored driving, is also unprotected against the sensor-related problems. Rotor position sensors not only reduce reliability but also increase cost and maintenance time. Positioning the sensor after maintenance or replacement requires special attention [2]. The position sensor requires more space inside or outside the motor and increases the cost [3]. Driving based on the motor model can be used to eliminate the sensor-related problems.

In recent years, a large academic community has expanded the application areas of artificial intelligence applications as much as possible. Artificial intelligence-based motor modeling approaches, sensorless control techniques, are used in SRM, one of the non-linear motors. In literature, models based on the change in the active phase current and using inductance and flux values in response to this change have been developed in studies with SRM [4]. Inductance, magnetic flux and active phase current are used to determine the rotor position with different approaches. Some of these are current waveform methods [3-5], high-frequency pulse injection methods (passive or active phase) [6-8], calculation methods for magnetic flux [9], [10], state observer methods (use non-linear) motor) [5,11,12] inductance variability methods [13-18] are mathematical model-based methods [1921]. With the pulse injection method, the currents applied to the passive phases tend to rise/fall according to the rotor position, and this behavior has been the most basic data used in sensorless control [7]. The pulses injected into the passive phases to obtain the inductance behavior of the motor, increase the noise of the system and can reach a level that may adversely affect the performance of the system in the phase commutation region [6-8]. In this approach, method-based noises that may adversely affect performance are kept at the lowest level.

The most difficult part of the pulse injection method is the decision of the activating/deactivating phase moment. In some studies, the next phase decision point was tried to be determined by using different threshold value approaches. These approaches are multiple thresholds [8], a constant threshold [24] or a dynamic threshold [18]. This study focuses on determining the rotor position of an SRM that can be used for high torque applications at a low constant speed by injecting high frequency pulses into the passive phases. Determining the rotor position at a certain point or region will facilitate the identification of a system that will adversely affect performance for a nonlinear motor. Instead, a machine learning algorithm trained with the current value generated by the high-frequency pulses injected

into the passive phases during its natural operation will allow a much more realistic system model to be trained. This approach provided a more natural and successful motor model. Induced currents measured from passive phases are used as input data in different machine learning classification algorithms. The active phase is estimated as output data. Success parameters were compared by calculating different performance criteria. The most successful model is proposed.

II. MATERIALS AND METHODS

A. Switched Reluctance Motor

Unlike motors used in industry, SRM does not contain permanent magnets and aluminum short-circuit bars. It is a simple motor with no transfer elements such as carbon brushes or rings. The rotational force is the same as the electromagnet principle and it is sufficient to energize the windings in the stator one after the other. The rotor of the SRM tends to align with the stator and rotor poles. The mathematical model of SRM consists of the electrical and the mechanical equation.

$$V = Rs.i + \frac{d\lambda(\theta,i)}{dt} \quad (1)$$

Equation (1) is defined for one phase of an SRM. In this voltage equation Rs refers to stator winding resistance, and i phase current and λ flux linkage for one phase.

$$\lambda = L(\theta,i).i \quad (2)$$

In (2) shows, Flux linkage λ is dependent on phase current i, and L is inductance directly related to the rotor position.

$$V = Rs.i + L(\theta,i)\frac{di}{dt} + \frac{dL(\theta,i)}{d\theta}.i\omega_m \quad (3)$$

Equation (3) consists of three parts from left to right, which is resistive voltage drop, inductive voltage drop, and electromotive force. The torque equation can be expressed in (4) for each phase.

$$T_e(\theta,i) = \frac{1}{2}i^2\frac{dL(\theta,i)}{d\theta} \quad (4)$$

The term i2 in the torque equation indicates that the torque is independent of the current direction. This feature allows the SRM to be controlled by both direct current and alternating current. It is also seen from the torque equation that the positive derivative of the inductance has the motor effect and the negative derivative can cause braking and reverse rotation. It is also seen from (4) that torque is the mathematical product of rotor position current.

These equations are used in computational mathematical modeling. However, the highly nonlinear nature of SRM requires some assumptions to be made such as mutual inductances are ignored and the motor is assumed to always operate in the linear region. It can be said that in the pulse injected passive phases, the diagnostic pulses produce very small negative torque due to the low duty cycle. For this reason, the effects of negative torque and diagnostic pulses are ignored.

The voltage induced by generation from diagnostic pulses applied to the passive phases to determine the rotor position can be approximately calculated from (5). V is the applied pulse voltage, L(θ) is the phase inductance of the idle phases, i_{peak} is the amplitude of the peak current value of the diagnostic pulse, and Δt is the period.

$$V \approx L(\theta)i_{peak}/\Delta t \quad (5)$$

The relationship between inductance, magnetic flux or induced voltage and rotor position has been used many times in the literature to determine rotor position. The easiest and fastest way to reveal the relationship between rotor position and induced current is to use passive phase currents. In this study, machine learning supervised classification algorithm based motor models were trained to determine rotor position and their performances were compared.

B. Preferred Machine Learning Algorithms

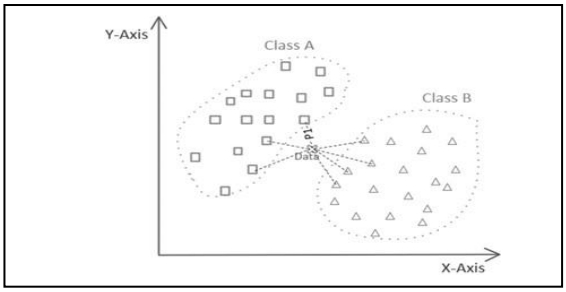
In modeling a 6/4 SRM with machine learning algorithms, the induced passive phase currents input is to form the energized active phase output parameter. Since predicting the active phase is a supervised classification problem, performance values are produced by training different algorithms. The success values obtained increased the accuracy success from 91% to 99.9% compared to the training with time series approach, instant passive phase currents. K-Nearest Neighbor algorithm and Decision Tree and Ensemble Bagged Tree algorithms were used.

C. K-Nearest Neighbors Algorithm

KNN (K-Nearest Neighbors) is a supervised machine learning algorithm that can be used for classification and regression problems. KNN would be very useful in nonlinear systems or real-world datasets that don't follow mathematical assumptions.

KNN has three steps for prediction or classification. The first step is to calculate the distance, the second step is to find the nearest neighbors and the final step is to vote for the class label. The KNN algorithm does not need a value outside the dataset or a random starting point [21].

Fig. 1. KNN Classification Algorithm



K is the number of nearest neighbors, which is the main decision factor. K is selected as in Fig. 1. For binary classification 7 it is usually a single number. The KNN algorithm first accepts the received point as the closest P_1. For the next estimate, the distance is measured with the next point. Until the new close distance is measured, the nearest measured distance is

P_1. Euclidean distance (6), Manhattan (7) and Minkowski (8) distance measurement methods can be used to find the closest distances. The distance between P_1, (x_1, y_1) and (x_2, y_2) can be measured by the following methods;

$$\text{Euclidean } P_1=\sqrt{(x_1-x_2)^2+(y_1-y_2)^2} \quad (6)$$

$$\text{Manhattan } P_1=|x_1-x_2|+|y_1-y_2| \quad (7)$$

$$\text{Minkowski } P_1=(|x_1-x_2|^p+|y_1-y_2|^p)^{1/p} \quad (8)$$

D. Decision and Ensemble Bagged Tree Algorithms

Decision tree algorithm is a method that solves problems using tree representation. Each branch separation corresponds to an attribute, each leaf to a class. The model training process starts from the root with a decision control. Attribute Selection Measures (ASM) splits records and creates smaller subsets. The tree formation process continues until the root, branches and finally the leaves. A general decision tree branching is shown in Fig. 2.

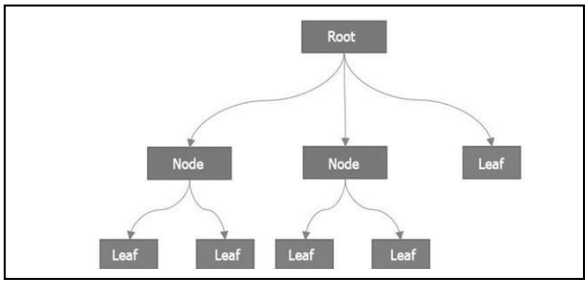


Fig. 2 Decision Tree algorithm with root, nodes and leaves

The source of the success of the decision tree algorithm comes from choosing which input data we will use in branching and determining its value in the next step. Simple structure, minimal branching and high success are expected from the algorithm. Typically, decision tree algorithms are unstable and ensemble method can make them stable and reliable [22].

E. Data Collection System with Pulse Injection

The asymmetric converter traditionally used in the SRM driver has two switches, high side and low side. The dual vphase switch must be driven by a more complex driver circuit with special snatch driver circuits. This complicates circuit and control, increases cost, and reduces reliability. In this setup, the power converter is simplified with star connection by connecting one switch for each phase. For current measurement, a shunt resistor was connected to each phase path and current measurement was performed (Fig. 3).

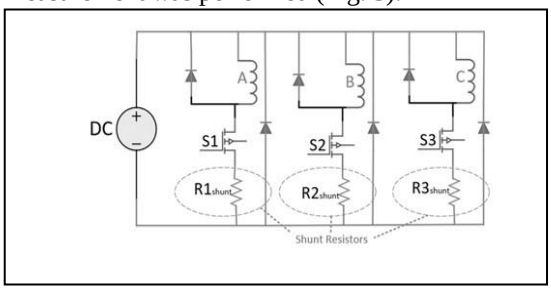


Fig. 3 Star connected power converter with phase current shunt resistor

Rotor position sensing signals are applied to the passive phases at low duty cycle without braking the motor, and the induced voltage is measured over shunt resistors. The commutation signal is applied to the active phase with a high duty cycle. The active phase current is controlled by the hysteresis controller so that the motor operates in the linear region. The active phase PWM duty cycle value is reset to prevent excessive current increase. In Fig. 4, the active phase is shown as Phase B. PWM pulses with high duty cycle applied to the S2 switch feed the phase. The rotor position determination signal to phases A and C is applied to switch S1 and S3 as a low duty cycle PWM pulse.

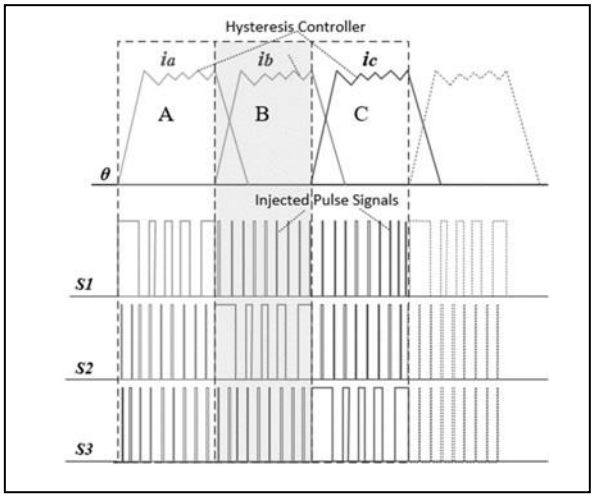


Fig. 4 Excited and idle phase signals with hysteresis control

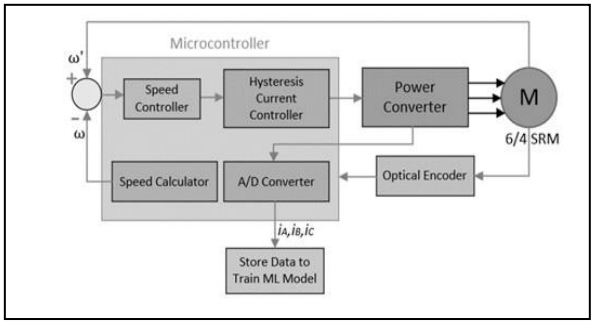


Fig. 5 Used SRM drive system to collect data

The drive system block diagram is given in Fig. 5. The motor was operated at constant speed and constant load, the currents for each phase were measured and saved to the computer. Time series approach is used to increase the system performance and train a more successful model than the fluctuating behavior of passive phase currents. In Fig. 6, input data, decision region and active phase regions are shown with time series approach.

III. Study and Results

A- Collecting Data

The data acquisition system consists of a 3-phase 6/4 SRM with an optical position sensor and driver circuit Fig. 7). The motor control is made by STM32F103C8

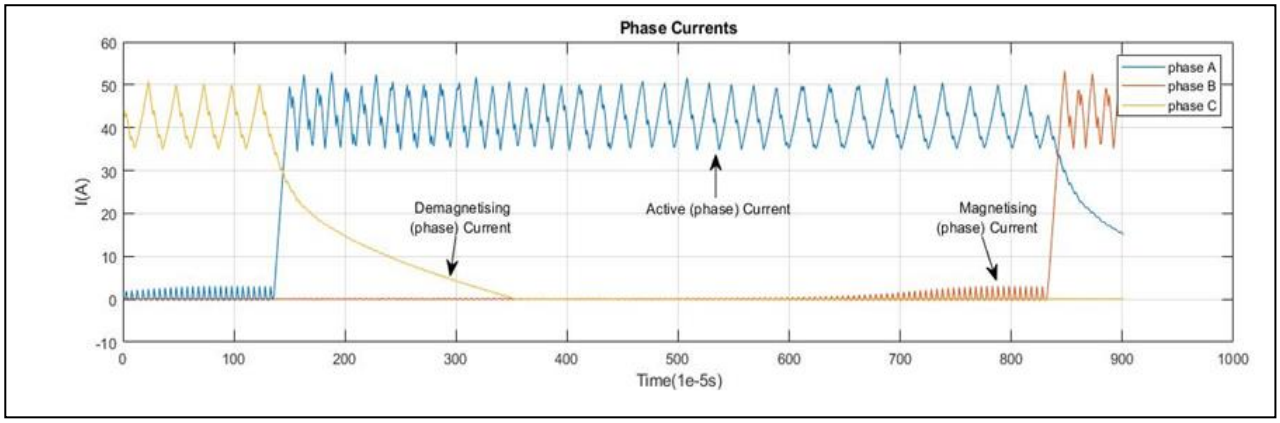


Fig. 6 Idle phase currents change according to the rotor position

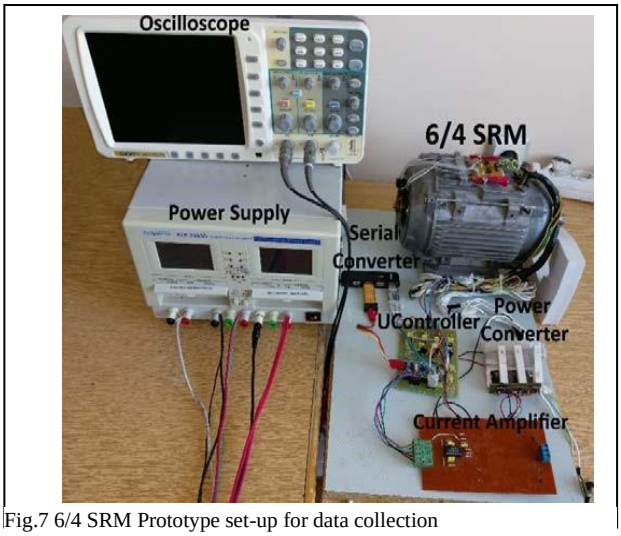


Fig.7 6/4 SRM Prototype set-up for data collection

running at 72 MHz with Cortex M3 core with ARM architecture.

Model training was carried out in the steps order of data preparation, algorithm selection, validation method selection, and parameter adjustment until the desired performance is achieved.

Previous	Next	Active
…	…	…
…	…	…
Ic	Ib	A
Ic(n+1)	Ib(n+1)	A
Ic(n+2)	Ib(n+2)	A
Ic(n+2)	Ib(n+2)	A
Ic(n+2)	Ib(n+2)	A
Ic(n+2)	Ib(n+2)	A
…	…	…
…	…	…
Ic(n+18)	Ib(n+18)	A
Ic(n+19)	Ib(n+19)	A
Ic(n+20)	Ib(n+20)	A
…	…	…

Fig.8 Raw Dataset

During the data preparation level, passive phase currents and active phase label formed the basic data set. Each row of the data matrix represents the passive phase currents and its label the active phase. The first column is the current of the previous (demagnetized) active phase, the second is the current of the next (to be magnetized) passive phase, and the last column is the label of the active phase. The active phase label can be A, B or C (Fig. 8).

prev20	next20	…	prev2	next2	prev1	next1	active
Ic(n+20)	Ib(n+20)	…	Ic(n+2)	Ib(n+2)	Ic(n+1)	Ib(n+1)	A
Ic(n+21)	Ib(n+21)	…	Ic(n+3)	Ib(n+3)	Ic(n+2)	Ib(n+2)	A
Ic(n+22)	Ib(n+22)	…	Ic(n+4)	Ib(n+4)	Ic(n+3)	Ib(n+3)	A
Ic(n+23)	Ib(n+23)	…	Ic(n+5)	Ib(n+5)	Ic(n+4)	Ib(n+4)	A
Ic(n+24)	Ib(n+24)	…	Ic(n+6)	Ib(n+6)	Ic(n+5)	Ib(n+5)	A
Ic(n+25)	Ib(n+25)	…	Ic(n+7)	Ib(n+7)	Ic(n+6)	Ib(n+6)	A
Ic(n+26)	Ib(n+26)	…	Ic(n+8)	Ib(n+8)	Ic(n+7)	Ib(n+7)	A
Ic(n+27)	Ib(n+27)	…	Ic(n+9)	Ib(n+9)	Ic(n+8)	Ib(n+8)	A

Fig.9 Time series converted dataset

Determining the rotor position from the instantaneous values of the passive phases has not been the suitable approach for an extremely non-linear high torque and current fluctuating SRM.

Although 90% accuracy was achieved in the models with instantaneous current measurements, it was observed that the at phase transition points are made errors. These errors are so crucial for driving.

Since the estimation of the phase transition point is the parameter that directly affects the performance of the motor, the margin of error should be extremely high and not in the phase transition regions. For this reason, the instantaneous passive phase currents as well as the 19 current values read first are arranged to form a line of the system input matrix as shown in Fig. 9. This approach, the time series approach is applied by associating the phase transition moment not only with the instantaneous but also with the previous 19 currents.

This approach is known as the time series method [23]. In Fig. 10, the graph shows the passive phase ADC values and the ADC values obtained in the experimental system, the excitation region used for the time series approximation values, and the excitation point.

The number of currents measured through a phase is related to the system speed and the number of measured data. The measured current number of the SRM during a phase transition is proportional to the microcontroller

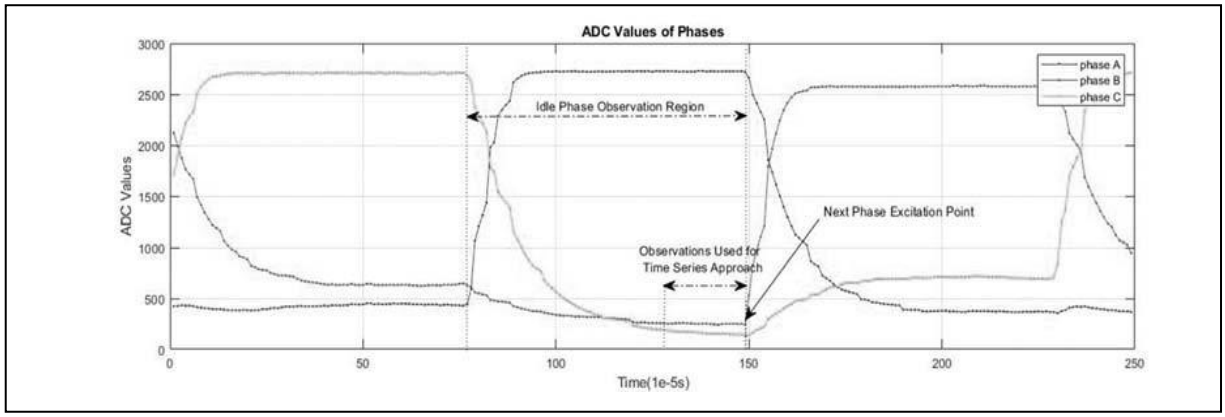


Fig.10 Phase currents obtained from the experimental setup

speed. As the speed increases, the number of measurements per phase transition will decrease. Modeling a system with less data means less data will be used in the model and lower accuracy and prediction. For this reason, successful systems should be trained by keeping the speed of the controller and the model size at an optimum level according to the motor speed.

B- Training and Comparison Performances of Models

The data obtained from the system operated at constant speed and load were rearranged with the time series approach. The most commonly used performance criteria of literature history classification algorithms are accuracy, precision, confusion matrix, memory usage, training time, prediction time, and interpretability. For real-time applications, prediction speed is also important.

Accuracy, the ratio of the predictions to all predictions, is the most frequently used performance testing parameter when testing the success of classification algorithms. Some key performance parameters of the dataset trained with machine learning classification algorithms are shown in Table 1.

TABLE I. ML CLASSIFICATION ALGORITHMS BASED ON THE ACCURACY

Algorithms	**Performance Parameters**		
Algorithms	***Accuracy (%)***	***Prediction Speed (obs/sec)***	***Training Time (sec)***
Coarse Tree	91.6	360,000	144.67
Medium Tree	98.7	380,000	117.30
Fine Tree	99.7	330,000	75.72
Tuned Fine Tree	99.9	190,000	385.48
Coarse KNN	99.7	74	20861.08
Cosine KNN	99.9	83	27040.5
Weighted KNN	99.9	75	35702.42
Medium KNN	99.9	75	19203.35
Fine KNN	100.0	60	11,725.76
Ensemble Boosted Tree	99.3	63,000	999.27
Ensemble Bagged Tree	99.9	39,000	648.71

Algorithms with an accuracy value of more than 90% are presented in Table 1. Algorithms with accuracy below 90% are not included in Table 1. These algorithms were: Linear and Quadratic Discriminant, Gaussian and Kernel Naive Bayes, Linear, Quadratic, Cubic and Gaussian Support Vector Machine, Weighted and Qubic KNN.

Although the accuracy rates in Table 1 provide a basic perspective on the success of the algorithms, they are insufficient to clearly demonstrate the performance. In addition to the accuracy value, sensitivity, precision and F1score values calculated from the confusion matrix are also used in the performance comparison.

The training dataset was separated from the training and validation set by the K-fold cross validation method. Data subsets were used for cross-validation for the trained model, taking k=5 for fold. Thus, all data are included in the training with different variations and the overfitting error of the model is prevented. For Fine KNN, the distances are equal and the distance measurement algorithm Euclid is chosen. Fine Tree model's accuracy value is 99.7%, maximum number of partitions: 100%, division criterion: Gini's diversity index and proxy decision division. The accuracy value of the fine tree model is increased to 99.9% by setting the maximum number of splits: 800, the split criterion: Maximum deviation reduction and proxy decision splitting: 30. The ensemble bagged tree was trained with the values of the ensemble method with an accuracy of 99.9%. Bag, plain type: Decision tree and number of learners: 30.

To determine the most successful model, the accuracy, recall, precision, specificity, F1-score and accuracy values of the most successful algorithms were compared in Table 2.

TABLE II. PERFORMANCE VALUES FOR ML ALGORITHMS

Algorithms	**Performance Parameters**					
	Recall	***Precision***	***Speci-ficity***	***F1-Score***	***Accuracy***	***Prediction Speed (obs/sec)***
Fine KNN	0.99954	0.99954	0.99977	0.99954	100.0	60.00
Ensemble Bagged Tree	0.99929	0.99928	0.99965	0.99929	99.9	39,000.00
Tuned Fine Tree	0.99913	0.99913	0.99957	0.99914	99.9	190,000.00

The successful model should have high accuracy and high prediction speed. In terms of performance criteria, although

Fine KNN model is the closest to natural motor behavior, it can be seen that the model is too slow to implement with 60 predictions per second. Ensemble bagged tree and Tuned Fine Tree models have very close performances. If the system is driven with the training speed environment ensemble bagged tree model can be accepted successful. But low estimation time reveal that the Tuned Fine Tree model can be used in systems with training speed and higher speeds.

IV. Conclusion and Discussion

In recent years, machine learning algorithms, which have found a place in artificial intelligence applications in many fields, have been used in different fields. Although there are different applications of the pulse injection methods in the literature, it is a new approach to use the instantaneous values of the passive phase currents in model training by increasing the features with the time series method.

Measuring individual phase currents is a simple modification on the driver circuit. For this reason, it does not require a base modification on mechanical or electronic from conventional drives. To train a driver motor model, it is sufficient to collect current data from a motor prototype with a position sensor. Thus, it has been shown that all motors with the same characteristics as the trained model can be successfully driven under training conditions. Due to the low processing capacity and memory of microcontrollers model training and running of machine learning algorithms cannot be performed on the controller yet.

On these days a limited amount of the models trained in the computer environment can be used on microcontrollers. Running computer-trained models only on computers is not suitable for real-time applications such as motor drive applications.

Although it is seen as a challenge that the trained models cannot be used in embedded systems, commercial companies aim to directly run artificial intelligence models in the microcontrollers they have developed with AI code. Hardware and software have started to be developed in this direction.

The developed model shows that non-linear electric motors can be driven with the driving parameters obtained under natural operating conditions with machine learning algorithms. With this approach, sensor-related errors of a motor-drive system can be predicted. Operating faults of the engine under critical conditions can be detected. Information about sensor errors and efficiency can be collected.

References

[1] Krishnan R (2001) Switched reluctance motor drives: modeling, simulation, analysis, design, and applications. CRC Press, London

[2] Miller TJE (2001) Electronic control of switched reluctance machines. Newnes, Australia

[3] Fard G (2009) Sensorless speed control of switched reluctance motor drive using the binary observer with online flux-linkage estimation. Iran J Electr Electron Eng 5(2):143–150

[4] Echenique E, Dixon J, Cárdenas R, Peña R (2009) Sensorless control for a switched reluctance wind generator, based on current slopes and neural networks. IEEE Trans Ind Electron 56(3):817–825

[5] Fahimi B, Emadi A, Sepe RB (2005) Four-quadrant position sensorless control in SRM drives over the entire speed range. IEEE Trans Power Electron 20(1):154–163

[6] Bateman CJ, Mecrow BC, Clothier AC, Acarnley PP, Tuftnell ND
(2010) Sensorless operation of an ultra-high-speed witched reluctance machine. IEEE Trans Ind Appl 46(6):2329–2337

[7] Khalil A et al (2007) Four-quadrant pulse injection and sliding mode observer-based sensorless operation of a switched reluctance machine over entire speed range including zero speed. IEEE Trans Ind Appl 43(3):714–723

[8] Pasquesoone G, Mikail R, Husain I (2011) Position estimation at starting and lower speed in three-phase switched reluctance machines using pulse injection and two thresholds. IEEE Trans Ind Appl 47(4):1724–1731

[9] Hu KW, Chen YY, Liaw CM (2015) A reversible position sensorless controlled switched-reluctance motor drive with adaptive and intuitive commutation tunings. IEEE Trans Power Electron 30(7):3781–3793

[10] Ofori E, Husain T, Sozer Y, Husain I (2015) A pulse-injectionbased sensorless position estimation method for a switched reluctance machine over a wide speed range. IEEE Trans Ind Appl 51(5):3867–
3876

[11] Paramasivam S, Vijayan S, Vasudevan M, Arumugam R, Krishnan R (2007) Real-time verification of AI based rotor position estimation techniques for a 6/4 pole switched reluctance motor drive. IEEE Trans Magn 43(7):3209–3222

[12] Gan C, Wu J, Yang S, Hu Y (2015) Phase current reconstruction of switched reluctance motors from DC-link current under double highfrequency pulses injection. IEEE Trans Ind Electron 62(5):3265–3276

[13] Hossain S, Husain I, Klode H, Omekanda A, Gopalakrishan S(2002) Four quadrant and zero speed sensorless control of a switched reluctance motor. In: 37th IAS annual meeting conference record of the 2002 IEEE industry applications conference (Cat. No. 02CH37344), vol 3, pp 1641–1646

[14] Khalil A et al (2005) A hybrid sensorless SRM drive with height- and six-switch converter topologies. IEEE Trans Ind Appl 41(6):1647–
1655

[15] Chen KH, Sun YK, Li TB (2016) Sensorless control of switched reluctance motor. Dianji yu Kongzhi Xuebao/Electric Mach Control 20(3):85–89

[16] Gao H, Salmasi FR, Ehsani M (2004) Inductance model-based sensorless control of the switched reluctance motor drive at low speed. IEEE Trans Power Electron 19(6):1568–1573

[17] Krishnamurthy M, Edrington CS, Fahimi B (2004) Prediction of rotor position at standstill and flying shaft conditions in switched reluctance machines. In: Nineteenth annual IEEE applied power electronics conference and exposition, 2004. APEC'04, vol 1, pp 537–544. IEEE

[18] Cai J, Deng Z (2012) Sensorless control of switched reluctance motor based on dynamic thresholds of phase inductance. ElectrPower Compon Syst 40(8):915–934

[19] Chang YT, Cheng KWE, Ho SL (2015) Type-V exponential regression for online sensorless position estimation of switched reluctance motor. IEEE/ASME Trans Mechatron 20(3):1351–1359

[20] Cai Jun, Deng Zhiquan (2011) Sensorless control of switched reluctance motor based on phase inductance vectors. IEEE Trans PowerElectron 27(7):3410–3423

[21] Thompson KR, Acarnley PP, French C (2000) Rotor position estimation in a switched reluctance drive using recursive least squares. IEEE Trans Ind Electron 47(2):368–379

[22] Ha K, Kim RY, Ramu R (2011) Position estimation in switched reluctance motor drives using the first switching harmonics through fourier series. IEEE Trans Ind Electron 58(12):5352–5360

[23] Zhang X, Li M, Zong XZhu, Cheng D (2017) Learning k for kNN classification. ACM Trans Intell Syst Technol 8(3):1–19

[24] Bontempi G, Ben Taieb S, Le Borgne YA (2013) Machine learning strategies for time series forecasting. In: Lecture notes in business information processing, LNBIP, vol 138, pp 62-70

Autonomous Pursuit-Evasion System in Friendly or Enemy Zones for UAVs

Enes Akdoğan
Department of Defence Electronics and Software
Baskent University
Ankara, Turkey
aeakdogan@gmail.com

Asst. Prof. Dr. Murat ÜÇÜNCÜ
Department of Electrical and Electronics Engineering
Baskent University
Ankara, Turkey
murat.ucuncu@yahoo.com.tr

***Abstract* — In this paper, an autonomous software for mini unmanned aerial vehicles is proposed to perform pursuit-evasion missions in combat zones, including friendly or enemy zones. The UAV mission area is divided into green, gray, and red areas according to the level of danger. The route to be followed while pursuing the enemy UAV in pursuit mission or escaping from the enemy UAV in evasion mission is decided according to the level of danger of these areas. The UAV platform, abbreviated as Turumtay, is used as the simulation environment (hardware-in-loop). The autonomous function of performing the tasks is achieved by using Reinforcement Learning and Q Learning. In Q learning, States, Actions, and Reward Tables are created according to the goal mission. Studies of these algorithms are examined by using example cases of three different cases for each mission type; a total of nine cases are analyzed. This software solution presents a concept design for autonomous missions, thanks to the advantages of the Q Learning algorithms. The developed Mission Control Software can easily be adapted to other problems by initializing proper reward tables and action sets for the desired mission types.**

Keywords* — *Reinforcement Learning, Q Learning, Pursuit-evasion, Autonomous Mini UAV, UAV Mission Control

I. INTRODUCTION

As a result of the widespread use of unmanned aerial vehicles, UAVs (Unmanned Aerial Vehicle) are actively used in many different missions in both civilian and military fields. It is anticipated that UAVs, used mainly for reconnaissance and surveillance, homeland security, and close air support at the tactical and strategic level in the military field, can also be used in air-to-air combats soon. Therefore, UAVs with technological equipment that can engage with other UAVs will be in an advantageous position. The ability to perform air-to-air combats autonomously will provide advantages in terms of both performance and time-criticality. In terms of Mini or Small UAVs, air-to-air combats are mainly focused on evasion and pursuit missions.

This paper analyzes the pursuit and evasion problem in UAV operation, and software is developed according to scenarios. The UAV operation area is divided into green, gray, and red zones. The Mission Control Software was designed to guide the UAV autonomously runs on a platform called "Turumtay". The software is responsible for pursuit and evasion missions while considering the safety of green and red zones. The problem focuses on one-to-one pursuit-evasion combat between Turumtay and Target UAV.

There are several pieces of research about pursuit-evasion problems and AI (Artificial Intelligence) algorithms like Q Learning [1], [2], [3]. However, in real scenarios, world maps are not entirely available for safety flights. There are enemy zones over which it is dangerous to fly due to enemy Air Defense Systems and other weapon threats. On the other hand, there are also ally zones that are safe for flights. In [4], [5], [6], [7], [8], they used avoidance zones or threat zones while creating the optimal path from one point to another point. In this study, these two problems are merged, and a new problem is defined as performing pursuit-evasion missions in an area that includes green, gray, and red zones. To solve this problem, Mission Control Software is developed, and that software is proposed as a concept solution for the pursuit-evasion problems with green, gray, and red zones.

II. PROBLEM DEFINITION

A. Agents

There are two agents in the environment. The first one is the platform abbreviated as Turumtay. The developed Mission Control software runs on Turumtay UAV, so Turumtay is the name of the friendly UAV. Target UAV is the other UAV in the environment. This UAV is the opponent of Turumtay. Turumtay and Target UAV pursuit each other or escape from each other.

B. Zone Types

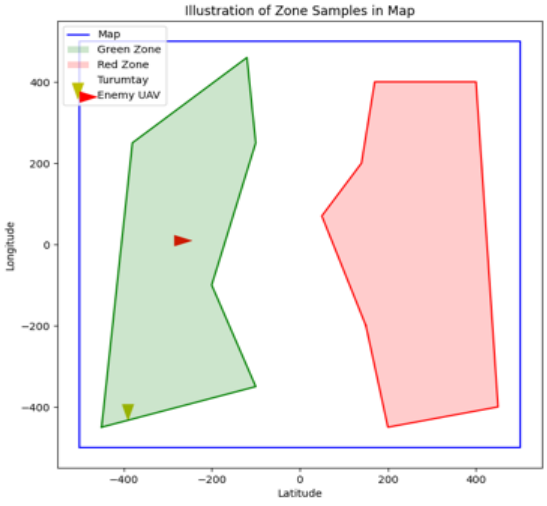


Fig. 1. Illustration of Zone Samples in Map

Three zones are defined to be available in the environment; Green, Gray, and Red zones depicted in Fig. 1. Gray zones are the rest of the points, which are neither green nor red zones. These green, gray and red zones symbolize ally, neutral, and enemy zones. Green zones are safe for flight. There are no known enemy threats, and the zone is out of range of enemy missiles. In Red zones, there are dangerous enemy threats to safe flight.

Zones are two dimensions shapes like polygons, and a set of points describes them. Sun [4] and You [5] represented the zones by using ellipses. However, in this paper, polygons represent zones like military zones. The advantage of using polygons is that there can be convex and concave shapes, which help determine the optimal path for the UAV to follow.

C. Mission Types

There are three types of missions: Patrol, Evasion, and Pursuit. Patrolling is a usual mission type for aircraft. In these missions, the plane goes between two points to detect anomalies in these regions. Patrolling missions are done in gray zones, so aircraft must stay in green or gray zones and not enter red zones.

In evasion missions, UAVs should not enter red zones and escape from Enemy Aspect Angle. If Turumtay is in the Field of View of Enemy, it is a potential danger. Therefore, Red Zones and Enemy Aspect Angles are considered critical dangers while escaping.

In Pursuit Mission, UAV prefers green zones or gray zones if possible and does not enter into red zones. Furthermore, Turumtay tries to lock on Enemy UAVs to keep Enemy UAVs in the Field of View of Turumtay.

III. SOFTWARE DEVELOPMENT

A. Reinforcement Learning and Q Learning

Reinforcement Learning is mainly based on Markov Decision Process (MDP). The Markov Decision Process defines states (S), actions (A), probabilities of actions (P_a), and rewards (R) to handle a problem. The process starts from the initial state. Then, a random act is applied, which changes the current state, and a reward of that state is gained. In Reinforcement Learning, the learning process is performed in a cycle of sense-act-learn [9]. On every step, reward according to that state is considered. MDP aims to maximize the final reward.

Reinforcement learning is a model-free learning model, which means that the Reinforcement Learning characteristics of the used model do not affect the learning process [10]. Thanks to this advantage, developed learning algorithms can be used within different platforms by minor changes [11].

Q Learning is one of the most used techniques in Reinforcement Learning which uses action-value functions [12]. The name of the Q Learning comes from the Q Table, which is used in learning algorithms to calculate the expected utility of actions [12]. Q Learning algorithms can work without prior knowledge about the results of actions, which makes solving real-world problems easier for agents.

B. Agent, Environment, Action, and Reward Modal

The agent performs an action at one state in the Agent, Environment, Action, and Reward modal. As the result of this action, the environment causes the state to transition with a reward.

1) State Set

States are denoted by S, and states are composed of X, Y, H, which are latitude positions, longitude positions, and heading angles. X, Y positions, and H heading angles are continuous elements. Therefore, these continuous spaces need to be divided into discrete spaces to solve this problem and apply algorithms.

$$output = \left(floor\left(\frac{input}{disc\_constant}\right) * \ disc\_constant\right) \quad (1)$$

Coordinates are composed of infinite points, but this infinity makes the problem impossible to solve [13]. The map and coordinate system are divided into equal-sized squares. The radius of these spheres is 20 meters, and the center of these spheres is considered one point in the coordinate system.

States also have an H heading angle element for UAVs. Exact heading angles are not crucial in pursuit or evasion; field of view is enough for these problems. So, heading angles are put into a process by dividing them by 45 degrees. To calculate discrete values for heading, Equation (1) is used when "disc_constant" is 45, and the input is the actual heading. So the output of the formula will be discrete heading angle.

2) Actions

Actions are denoted by A, and there are three possible actions that Turumtay can do during the transition from one state to another. These actions are moving forward left, forward center, and forward right. Turning heading angle causes these movements by turning 45 degrees left, maintaining heading, and turning right by 45 degrees. These actions are illustrated in Fig. 2.

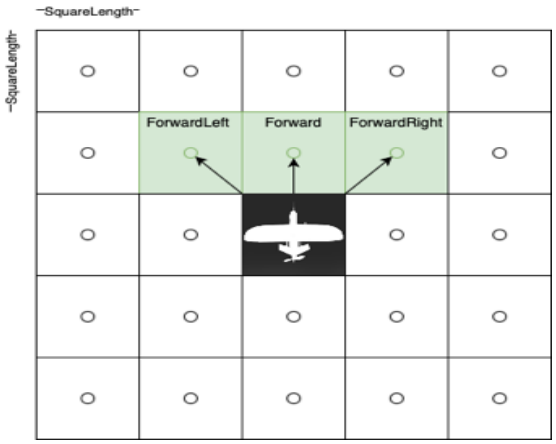


Fig. 2. Illustration of Actions

3) Rewards

In Q Learning problems, the agent aims to gain maximum cumulative reward until it reaches the target state, so constructing proper reward functions is critical for the agent's performance in the mission [6]. Rewards depend on the type of mission. Each mission type has different reward functions regarding its goals.

To calculate rewards of states, the distance between Turumtay and Target UAV, Turumtay's Aspect Angle, and Enemy UAV's Aspect Angle is used.

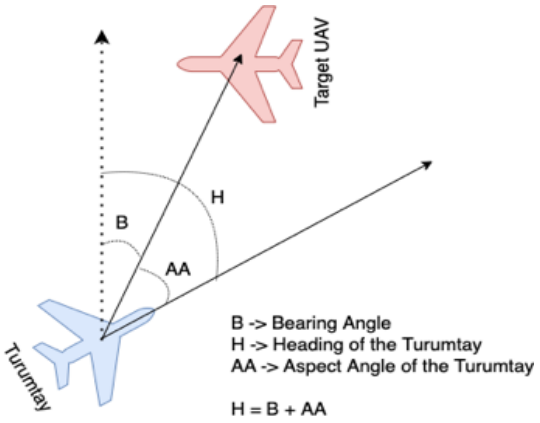


Fig. 3. Illustration of the Bearing and Aspect Angle

Aspect Angle is like bearing angle. However, the aspect angle heading of the state is also considered because this angle shows the field of view of the UAV. Aspect Angle is calculated by using bearing angle, and the only difference is that bearing angle is subtracted from the heading angle of the UAV. There is an illustration of the bearing angle and the aspect angle of the Turumtay in Fig. 3. There are four Aspect Angle categories; Forward, Left, Right, and Back. Forward covers "-45 < A < 45", Left covers "-135 <= AA (Aspect Angle) <= -45", Right covers "45 <= AA <= 135", and other angles are in Back category.

The distance is also continuous so, distance is grouped as Too Close, Close, Far, and Too Far. Too Close covers a distance smaller than 50 m, and Close covers a distance smaller than 200 m. Distances between 200 meters and 750 meters are included in the Far group. Lastly, Too Far has distances over 750 meters.

Reward functions are created according to specific requirements of missions. So gained rewards are determined in order to satisfy mission requirements. Numerical values such of specific rewards are not absolute values, they are only imply the utility of that particular situation. For instance, agent gain -9999 rewards when it is out of the map, -9999 is not absolute number it may be -5000, -8000 or -20000. The intention is setting huge penalty in order to prevent agent to reach map boundaries.

In Patrolling mission, the main goal is to reach the waypoint. In addition to this, avoiding entering red zones is critical. While creating the path, Turumtay should prefer to stay in the green zone rather than gray zones, and it should choose the shortest route. Details of the reward function for patrol missions can be seen in TABLE I. If Turumtay exits from the map's limits, it is penalized with a -9999 point. If it reaches the goal state, it receives 3333 points because, in the patrol mission, there are three sub-routes which are from initial position to first patrol point, from the first point to second point, and from the second point to first point. So, to make a maximum point to 10000, the rewards of 3 sub-missions are equal to 3333. If it chooses red zones, it takes -1000 points. On the other hand, the penalty is only -1 in the green zone and is -10 in gray zones. All green, red, and gray points have negative reward points to make the path shorter and prevent unnecessary longer ways.

TABLE I. TABLE OF REWARDS IN PATROL MISSIONS

Reward	State Condition
-9999	State out of the map
-1	State in Green Zone
-1000	State in Red Zone
-10	State in Gray Zone
3333	State is closer than 20 meters to Patrol Point

While performing an escape mission, Turumtay tries to escape to a safe zone considering green, red zones information and target UAV's position. Turumtay prefers shorter paths and keeps itself away from the target UAV while escaping. Implementation of the reward function for the escape mission can be seen in TABLE II. States out of the map are penalized with -9999 points, and states in the red zone are penalized with -1000 points. If the distance between Turumtay and Target UAV is Too Close. A reward would be -200 and if the distance is Too Close, and if the distance is Close, reward would be -100 points. In addition to these penalties, if Turumtay is in the field of view of the Target UAV and the distance is smaller than Close, the reward would be -400. The reward of entering the green zone while distance is Far or Too Far is 10000, and it is the mission's primary goal. In other cases, Turumtay gets a -1 point as a reward to prevent unnecessary movements that cause longer paths.

TABLE II. TABLE OF REWARDS IN EVASION MISSIONS

Reward	Condition
-9999	State out of the map
-1	State in Green Zone
-1000	State in Red Zone
-10	State in Gray Zone
-200	Distance between Turumtay and Enemy UAV is smaller than *Too Close*
-100	Distance between Turumtay and Enemy UAV is smaller than *Close*
-400	Distance between Turumtay and Enemy UAV is smaller than *Close*, and Enemy's Aspect Angle is *Forward*
10000	Turumtay in *Green Zone* and distance between Turumtay and Enemy UAV is larger than *Close*

Pursuit missions are the opposite of escape missions. TABLE III. summarizes the reward function for pursuit missions. Like Patrolling and Escaping missions, states located out of the map are penalized with -9999 points, and states in the red zone are penalized with -1000 points. If Turumtay is Close to Target UAV and Target UAV is in the field of Turumtay and Turumtay is not in the field view of Target UAV, the reward is 10000 points because these states are desired states in the pursuit missions.

TABLE III. TABLE OF REWARDS IN PURSUIT MISSIONS

Reward	Condition
-9999	State out of the map
-1	State in Green Zone
-1000	State in Red Zone
-10	State in Gray Zone
10000	Distance between Turumtay and Enemy UAV is equal to *Close*, and Turumtay's AA is Forward, and Enemy's AA is Back

IV. RESULTS AND EVALUATION

In Reinforcement Learning problems, performance criteria are defined by using reward tables, so each problem has different success metrics according to their reward functions [3], [5], [14]. In this study, problem definition covers these two types of problems: path planning regarding zone types and pursuit-evasion missions. So, the result is not distinct as a win or lose because of the different rewards on different missions. The mission is successful as gained total reward.

To evaluate the algorithm's performance, 500 random cases are created for each mission type and simulated these cases.

In patrol missions, Turumtay gained 9636 average rewards out of 10000. The results of earned rewards on 500 patrol mission cases are presented in Fig. 4. Individual rewards are spread between 9250 and 9750 points, and the mean of 500 cases is 9636.

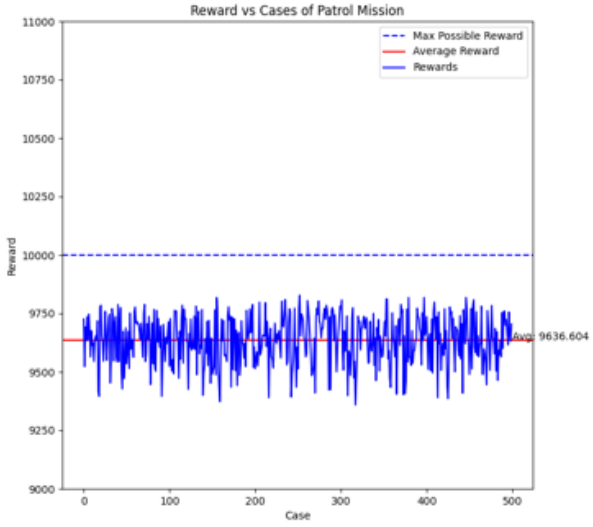


Fig. 4. Reward vs. Cases Chart of Patrol Missions

Turumtay gained 9939 reward points out of 10000 in evasion missions, and rewards are between 9750 and 10000 generally. In these cases, the initial position of the Turumtay effect gained reward. So, it is inevitable that when Turumtay is located too far from Green Zone, the accumulated reward can be smaller than 9000. However, the number of these cases is negligible, as can be shown in Fig. 5. The mean of reward in these 500 cases of Escape mission is 9939 out of 1000

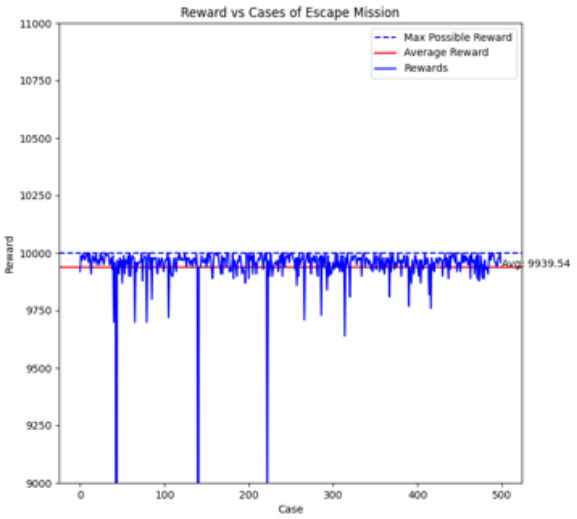


Fig. 5. Reward vs. Cases Chart of Escape Missions

The mean value of reward gained in 500 cases of pursuit mission is 9739. Rewards in these cases are between 9500 and 10000 for most of the cases, as can be seen from Fig. 6. 9739 out of 10000 points on average show that mission control can find a path to come close Enemy UAV from behind and keep Enemy in Turumtay's field of view in each case of pursuit missions.

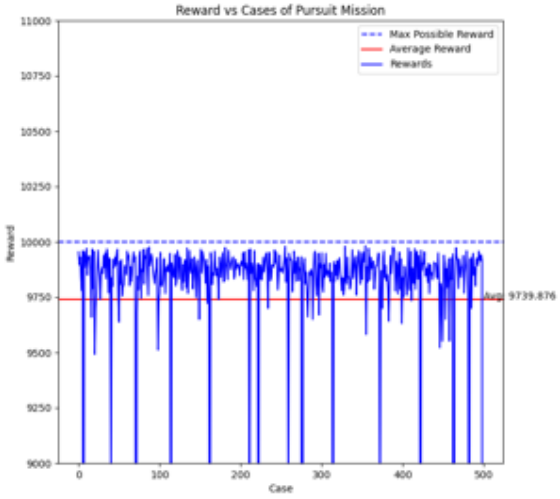


Fig. 6. Reward vs. Cases Chart of Pursuit Missions

V. CONCLUSION

In summary, this paper proposed a software as a solution for pursuit-evasion problems of Mini UAVs in areas that include green, gray, and neutral zones. Mission Control Software used Reinforcement Learning methods and Q Learning algorithms to find optimal paths while performing missions. In Q learning, States, Actions, and Reward Tables are created according to the goal mission.

The results of developed software in this paper are analyzed. The findings of this study show that autonomous software to perform pursuit-evasion missions in combat areas,

including friendly zones or enemy zones, can be developed using Reinforcement Learning and Q Learning methods.

TABLE IV. AVERAGE REWARDS FOR EACH MISSION

Mission	Average Reward (out of 10000)
Patrol	9636
Evasion	9939
Pursuit	9739

To assess the performance of the Mission Control, 500 random cases were simulated for each mission: patrol, evasion, and pursuit. The average reward values of these cases are listed in TABLE IV. In patrol missions, the average reward value is 9636, the reward value in evasion missions is 9939, and the reward value in pursuit missions is 9739 out of 10000.

REFERENCES

[1] J. McGrew, J. How, L. Bush, B. Williams and N. Roy, "Air-Combat Strategy Using Approximate Dynamic Programming," *Journal of Guidance, Control, and Dynamics*, 2010.

[2] X. Ma, L. Xia and Q. Zhao, "Air-Combat Strategy Using Deep Q-Learning," *2018 Chinese Automation Congress (CAC)*, pp. 3952-3957, 2018.

[3] B. Vlahov, E. Squires, L. Strickland and C. Pippin, "On Developing a UAV Pursuit-Evasion Policy Using Reinforcement Learning," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2018.

[4] C. Sun, Y.-C. Liu, R. Dai and D. Grymin, "Two Approaches for Path Planning of Unmanned Aerial Vehicles with Avoidance Zones," in *Journal of Guidance, Control, and Dynamics*, 2017.

[5] S. You, C. Wan and R. Dai, "Iterative Learning Optimization for UAV Path Planning with Avoidance Zones," in *2019 American Control Conference (ACC)*, 2019.

[6] C. Yan and X. Xiang, "A Path Planning Algorithm for UAV Based on Improved Q-Learning," in *2018 2nd International Conference on Robotics and Automation Sciences (ICRAS)*, 2018.

[7] A. Gupta, A. K. Gupta, C. Bocaniala and V. V. S. S. Sastry, "Avoidance of threat zone by UAV for automated navigation," in *2008 Annual IEEE India Conference*, 2008.

[8] B. Zhou, W. Wang, Z. Wang and B. Ding, "Neural Q Learning Algorithm based UAV Obstacle Avoidance," in *2018 IEEE CSAA Guidance, Navigation and Control Conference (CGNCC)*, 2018.

[9] N. Imanberdiyev, C. Fu, E. Kayacan and I. Chen, "Autonomous navigation of UAV by using real-time model-based reinforcement learning," in *14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2016.

[10] T. Zhou, M. Chen and J. Zou, "Data Fusion of Air Combat Based on Reinforcement Learning," in *2019 IEEE 4th International Conference on Advanced Robotics and Mechatronics (ICARM)*, 2019.

[11] S. Tosunoğlu, "Autopilot System And Ground Station Software For UAV's," Istanbul Technical University Department Of Aeronautics And Astronautics, 2013.

[12] S.Manju and M.Punithavalli, "An Analysis of Q-Learning Algorithms with Strategies of Reward Function," *International Journal on Computer Science and Engineering (IJCSE)*, vol. 3, no. 2, pp. 814-820, 2011.

[13] H. X. Pham, H. M. La, D. Feil-Seifer and L. V. Nguyen, "Reinforcement Learning for Autonomous UAV Navigation Using Function Approximation," *2018 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pp. 1-6, 2018.

[14] A. Toubman, J. J. Roessingh, P. Spronck, A. Plaat and J. V. D. Herik, "Rewarding Air Combat Behavior in Training Simulations," in *2015 IEEE International Conference on Systems, Man, and Cybernetics*, Hong Kong, China, 2015.

Honeycomb Cell Detection and Classification by Self-Constructed Convolutional Neural Network

Maliha Farahmand
Department of Computer Engineering Faculty of Technology, Selçuk University
Konya, Turkey
ORCID ID 0000-0001-9887-0618

Dr Mehmet Akif Şahman
Department of Electrical and Electronics Engineering, Faculty of Technology, Selçuk University
Konya, Turkey
ORCID ID 0000-0002-1718-3777

***Abstract* – Harvesting the honeycomb with the right methods has great importance for beekeeping activities and the country's economy. Failure to use the right methods in honey harvest will adversely affect both the production quantity and quality of honey. Unconscious beekeeping activities prevent bees, which play a major role in the continuation of natural life, from continuing their generation. In this study, honeycomb cells were detected and classified using the deep learning method. As an example, a honeycomb dataset consisting of 7(empty cells (0), closed cells containing larva (1), open cells containing larva (2), closed cells containing honey, open cells containing honey (4), cells containing Pollen (5) and unused cells (6)) classes and a total of 103451 training and 25863 test images were created. Classification is done using self-structured CNN(SSCNN) architectures. As a result of the classification process, 94% accuracy for training and 93% accuracy for validation were obtained.**

Keywords— classification, honeycomb, deep learning, convolution neural network

I. INTRODUCTION

Beekeeping has been historically practiced in various places as an activity related to the production of pollinators as well as the local food culture. For example, in countries such as Japan, Korea and India, honey is an important element of food culture [1]. Around 35% of the western human diet benefits directly or indirectly from honeybee pollination. Therefore, the protection of beekeeping and honey production can greatly contribute to the maintenance of food culture and Honeybees play a vital role in modern agriculture [2]. They have antimicrobial, antiviral, antibacterial and antiparasitic properties due to their minerals, vitamins and enzymes. Honeybees, honey, beeswax, royal jelly, bee venom, pollen and propolis are extremely valuable for human health and nutrition. Therefore, beekeeping has both strategic and economic importance. Bees and other pollinators such as butterflies, bats and flies are increasingly threatened by human activities. But pollination is a fundamental process for the survival of our ecosystems. About 90% of the world's wild flowering plant species, along with more than 75% of the world's food crops and 35% of global farmland, depend entirely or at least in part on animal pollination. Pollinators not only directly contribute to food security but are also key to preserving biodiversity. Bees and other pollinators are declining in abundance in many parts of the world, largely due to intensive farming practices, monocropping, overuse of agrochemicals, and high temperatures associated with climate change. Apart from the global deaths of bees, there are also losses in bee colonies due to beekeepers' wrong approaches, not taking timely precautions and lack of information. The biggest cause of these deaths is the weakening of bee colonies. When bee colonies are weakened, bees die due to viruses (The Sacbrood Virus (SBV), the Black Queen Cell Virus (BQCV) and Chronic Bee Paralysis Virus (CBPV) etc.) and harmful species (Nosemosis, Varroa mites, etc.), their offspring do not develop naturally and they cannot resist predators (wasp, hornet, etc.). Honey filtering process to ensure the protection of the honeycomb is the most important step of receiving secret honeycomb that created by bees. The technologies used currently cause all honey to be filtered, regardless of the offspring in the comb, causing the death of the offspring or renounced the honey to protect the offspring in the combs. Beekeepers gain more efficiency by ignoring the offspring, but they cause the bee colony to weaken, thus making it unable to be protected from external factors (disease or plunder). The short-term profit-making efforts of beekeepers harm them in the long-term. When beekeepers do not harvest the honeycombs with brood on them, they are faced with a decrease in the amount to be obtained. In case of waiving the honey in the honeycombs with offspring, approximately 10% of honey is lost. In order to prevent these losses in honey harvest in beekeeping, the cells in the honeycomb must be determined correctly. In this study, the problem of correct detection of cells in the honeycomb is considered as a classification problem. As a result of the research, seven class labels were found. These classes are labeled as empty cells (0), closed cells containing larva (1), open cells containing larva (2), closed cells containing honey (3), open cells containing honey (4), cells containing pollen (5) and unused cells (6). A sample study data set was created with these labeled data, and detection and classification were performed with a deep learning approach.

II. MATERIALS AND METHOD

A total of 19 standard Langstroth beehives with different properties were used for this study. Since each honeycomb has two faces, a total of 38 honeycomb images have been studied. The reason for working with honeycombs with different characteristics is that the honeycombs have different characteristics and increase the comprehensiveness of the study. These characteristic features include the type of bee, the region of the bee, the nectar it collects for honey (pine, fig, sunflower, corn, flower, etc.), These factors change the color texture of the honeycombs. In this study, the BASLER acA2500-14uc field scanning camera was used to obtain images. The BASLER acA2500-14uc field scanning camera we used in the project provides images with a maximum screen resolution of 2590x1942 and a minimum of 64x64 pixels. In addition, the gain value can vary between 0-23.7. A cabin was prepared where the noises in the external environment can be prevented while the images are being

taken, and a light source is used frame images were taken at 2590x1940 pixels, but the working area is limited to the part where the honeycomb is located. The dimensions of the honeycomb region were determined as 1162x574 pixels (Fig. 1)

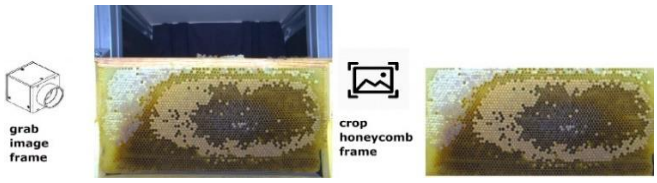


Fig. 1. Phase of taking honeycomb image

In the examinations, it was seen that the width of each cell in the honeycomb was 14 pixels. Therefore, the honeycomb with the size of 1162x574 pixels is divided into small pieces of 14x14 pixels. Thus, a total of 3403 pieces of painting were obtained from a honeycomb image, with 83 columns and 41 row. Images are taken from TUBITAK 1512 Techno-Enterprise Capital Support Program project numbered 2170060 (Fig. 2).

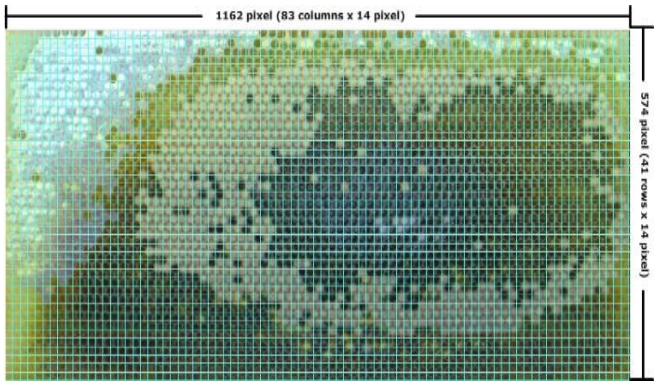


Fig. 2. Breaking into pieces of comb images

Cells in the comb can have different states. As mentioned before, open cell containing some empty cells (0), closed cells containing larva (1), open cells containing larva (2), closed cells containing honey (3), open cells containing honey (4), cells containing pollen (5) and unused cells (6) were determined as (Fig. 3).

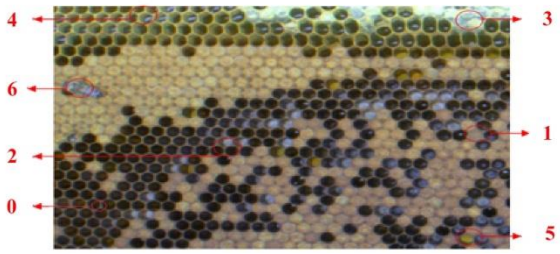


Fig. 3. View of the cell species

In this study, current situations are collected in seven main classes. One of the reasons is that beekeepers should not use open cell combs with many larva when harvesting honey. In case of harvesting combs with offspring in open cells with larva, both the offspring form the next generation will be lost and these offspring will be mixed with the filtered honey and the homogeneity of the honey will deteriorate. The other reason is that by detecting the seven classes, we process the cells that need to be processed and not waste time for other cells. In this way, the offspring are not harmed, and the homogeneity of the honey is not disturbed. Because the honeycomb does not have a standard design, the grids created for image selection do not overlap exactly on the cell. If it is more than 50% in any class in each cell, that cell carries the characteristics of that class (Fig. 4).

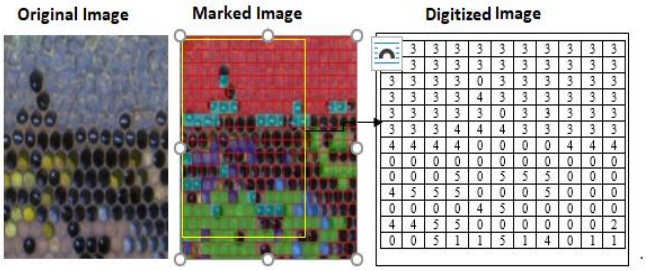


Fig. 4. Phase of digitization of image

14x14 pixel image with 3403 marks was obtained from each selected honeycomb. Total of 38x3403 = 129314 identified image pieces were obtained from 38 honeycomb images. These images, 68476 empty cells (0), 26554 closed cells containing larva (1), 1832 open cells containing larva (2), 22042 closed cells containing honey (3), 7338 open cells containing honey (4), 2144 cells containing pollen (5) and 928 unused cells (6).

III. Deep Learning

Deep learning can be defined as a class of machine learning techniques that leverages multiple layers of nonlinear information processing for supervised or unsupervised feature extraction, transformation, pattern analysis, and classification. Deep learning is a set of computational models consisting of multiple processing layers that enable learning the properties of data with more than one abstraction (separation) level [3]. The basic concept in deep learning algorithms is the automatic extraction of features from data. Deep learning algorithms are a good research method for automatically extracting complex data (features) at high abstract levels. These algorithms create a layered learning and data architecture in which higher-level (more abstract) features are defined as lower-level (less abstract) features [4].

A. Convolutional Neural Networks

It is a neural network made up of nerve cells that communicate with each other. Each neuron has weight values, and by updating these weights and training the network, the model produces accurate results [5]. It has been shown to have great potential for classifying images and recognizing objects in images [6]. The main purpose is to train the network and determine the filter weights.

B. Artificial neural networks

Artificial neural networks are a type of machine learning inspired by the complex functioning of the human brain, where hundreds billions of interconnected neural cells process information in parallel [7]. In such networks, it has an extraordinary ability to predict, classify and match data. These networks are segmented in different ways, the simplest and most widely used is the multilayer perceptron neural network. These networks, following mathematical relationships and patterns of human brain to understand the

complex relationship between the input data to accurately predict the output of small errors [8].

C. Regularization techniques

Commonly used neural network editing techniques are explained in detail in this section.

- **Data augmentation** artificially increases the variety and number of training samples by performing random transformations on existing images to generate a set of new variables without changing the meaning of the data. Flipping, rotating, adding noise are commonly used data augmentation techniques. Data augmentation is used to avoid overfitting and is particularly useful when the training dataset is relatively small [9].
- **Dropout** method to prevent neurons from overfitting certain features of the data. During the training phase, a dropout layer put out of action each neuron in that layer with a probability p where p is $0 < p < 1$. Deep neural networks are universal function estimators, so these techniques aim to reduce the overfitting of training data to increase the generalization capacity of the neural network [9].
- **Early stopping** is a technique to reduce overfitting by using some of the training dataset as a validation set. The training process does not involve this data. If the error of the validation set reaches a certain value, the training is stopped during the training step It can be revealing an overfitting is exist in current neural network for the training data [10].

IV. SSCNN (Self-Structure CNN)

SSCNN architecture is used for detection and classification of Honeycomb. In this architecture we use 6 convolutional layers, Maxpooling layers, Dropout layers, Relu, Softmax activation layers in Fig. 5 explain more details.

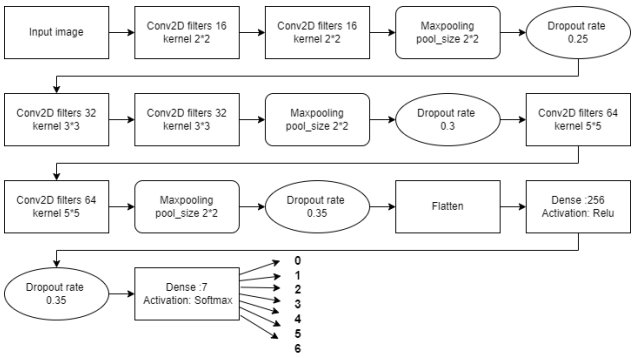


Fig. 5. Self-Constructed Convolutional Neural Network

V. Performance Metrics

In a classification problem, the evaluation metric was used in two phases, the training phase (learning process) and the testing phase. In the training phase, the evaluation metric was used to optimize the classification algorithm. In other words, the evaluation metric was used as a discriminant to distinguish and to select the optimal solution that could produce a more accurate estimate of the future evaluation of a given classifier. Meanwhile, in the testing phase, the evaluation metric was used as an evaluator to measure the effectiveness of the generated classifier when tested with test data [11].

A. Classification criteria

At the classification step, Criteria are used to find the criteria such as distinguishing the most appropriate, choosing the classification method, determining the significance between the classes, and estimating. Table I present the criteria used in the study to compare classification performances [12].

TABLE I. Metrics for Classification Evaluations

Criterion	Equation	Explanation
Accuracy	$\frac{(TP+TN)}{(TP+FP+TN+FN)}$	Correct estimation of the total number of samples
Sensitivity	$\frac{TP}{(TP+FN)}$	Positive samples that are properly classified
Precision	$\frac{TP}{(TP+FP)}$	Positive samples that are correctly predicted from all the predicted samples in a positive class
F-score	$\frac{(2*TP)}{(2*TP+FP+FN)}$	Correct optimization of the system

VI. Experimental Results

In this study, all experiments result was using a PC with i9 processor, Gtx 960m, NVIDIA GeForce 2080 Ti 9GB GPU card, 128GB RAM and Windows 10 pro-64-bits. In addition, Python 3.9 software was used for performance evaluation of the network's architecture. Moreover, deep learning libraries like TensorFlow, Keras has utilized. Which provides a framework for designing and implementing CNNs, where programs and graphics help visualize network activation and monitor network training progress.

TABLE II CLASSIFICATION RESULTS

Classes	Precision	Sensitivity	F-score	Support
Class 0	0.89	0.97	0.93	5352
Class 1	0.97	0.97	0.97	5310
Class 2	0.87	0.69	0.77	366
Class 3	0.97	0.94	0.96	4408
Class 4	0.93	0.88	0.91	1467
Class 5	0.96	0.50	0.66	428
Class 6	0.84	0.78	0.81	185
Accuracy	0.94			17516
Macro Average	0.92	0.82	0.86	17516
Weighted Average	0.94	0.94	0.94	17516

The dataset used in this study comprises 129314 images consisting of 7 (0-6) classes were marked from a total of 38 honeycomb images. 103455 of these images were used for training and 25859 for testing it means 80% of these images were used for training and 20% for testing. The training and test images were randomly selected, and we used SSCNN architectures in 30 epochs applied on the data set were carried out and the results of the application were analyzed. From the classification results, accuracy, sensitivity, f-score, precision of each class is given in Table II.

As seen in Fig. 6 the graphs of the loss and Accuracy functions obtained during the training and validation stages of the models are as follows.

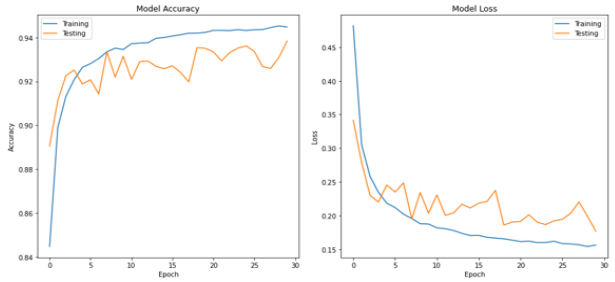


Fig. 6. Training and validation accuracy and loss

One of the most used success evaluation methods in machine learning applications is Roc curve. The graphs show how the model is successful. Call the area under the curve AUC. The Wide of this area explain the better accuracy of the model. It is given in Fig. 7.

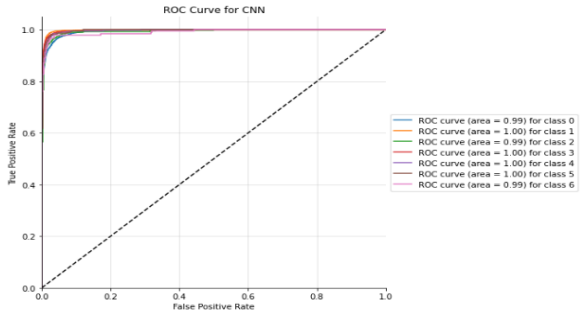


Fig. 7. Roc curve

The accuracy rate can evaluate the classification ability of the model, but specific details cannot be reflected. Confusion matrix, also known as error matrix in the field of machine learning and especially in statistical classification problem. It is a special table layout that allows visualization of the performance of an algorithm. The confusion matrix clearly shows the prediction details of each category when the classification model makes predictions, comparing the predicted outcome with the actual value. Confusion matrix of honeycomb classification model is shown in Fig.8.

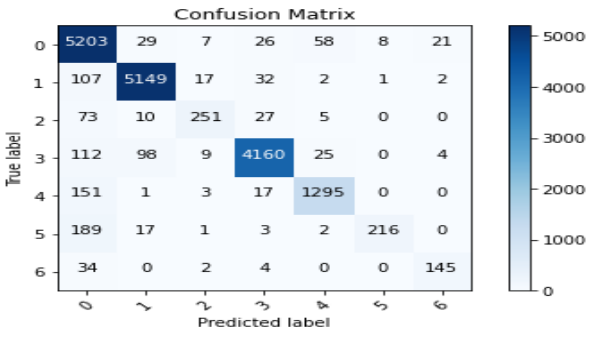


Fig. 8. Confusion Matrix

VII. Conclusion

In this study, basic artificial intelligence techniques used for classification are mentioned and information is provided on the criteria that can be used to measure the performance of classifiers. In this study, honeycomb images from the completed TÜBİTAK-1512 project were transferred to digital media by increasing the number of classes from two to seven. Newly labeled dataset was measured using deep learning convolutional neural network, with different metrics such as f1-score, precision, sensitivity and roc curve. Depending on these purposes, python 3.9 software, TensorFlow, Keras deep learning libraries were utilized on the experimental computer. After the experimental studies, 94% training accuracy and 93% validation accuracy were obtained. Due to the data set having very little data in some classes, the accuracy rate in these classes became lower than in others.

REFERENCES

[1] Kohsaka, R., M.S. Park, and Y. Uchiyama, *Beekeeping and honey production in Japan and South Korea: past and present.* Journal of Ethnic Foods, 2017. **4**(2): p. 72-79.

[2] Tarpy, D.R., D. vanEngelsdorp, and J.S. Pettis, *Genetic diversity affects colony survivorship in commercial honey bee colonies.* Naturwissenschaften, 2013. **100**(8): p. 723-728.

[3] LeCun, Y., Y. Bengio, and G. Hinton, *Deep learning.* Nature, 2015. **521**(7553): p. 436-444.

[4] Ünaldı, I., *Breast cancer histopathological image classification with convolutional neural networks models.* Higher Education Council Publication and Documentation Department National Thesis Centeri.

[5] Emrah Çevik , K.Z., *Classification of Skin Lesions in Dermatoscopic Images with Deep Convolution Network.* 2019.

[6] Ba, W., et al., *Diagnostic assessment of deep learning for melanocytic lesions using whole-slide pathological images.* Translational Oncology, 2021. **14**(9): p. 101161.

[7] Reese, K.M., *Deep learning artificial neural networks for non-destructive archaeological site dating.* Journal of Archaeological Science, 2021. **132**: p. 105413.

[8] Chen, H., et al., *Optimization of inflow performance relationship curves for an oil reservoir by genetic algorithm coupled with artificial neural-intelligence networks.* Energy Reports, 2021. **7**: p. 3116-3124.

[9] Ada, S.E., *Transfer learning for continuous control.* Higher Education Council Publication and Documentation Department National Thesis Center, 2019.

[10] Ergin, F., *Deep earning analysis in dermoscopy images.* Higher Education Council Publication and Documentation Department National Thesis Centeri, 2020.

[11] Hossin, M. and M.N. Sulaiman, *A review on evaluation metrics for data classification evaluations.* International journal of data mining & knowledge management process, 2015. **5**(2): p. 1.

[12] Birkan Büyükarıkan, E.Ü., *Fruit Classification with Convolution Neural Network Models using Illumination Attribute.* Uludag University Journal of Engineering Faculty, 2020.

Shilling Attack Detection for Recommender Systems with Quality Tools

Halil İbrahim AYAZ
Department of Industrial Engineering
Necmettin Erbakan University
Konya, Turkey
hiayaz@erbakan.edu.tr

Zehra KAMIŞLI ÖZTÜRK
Department of Industrial Engineering
Eskişehir Technical University
Eskişehir, Turkey
zkamisli@eskisehir.edu.tr

***Abstract*— Recommender systems have an important role to overcome information overload problems. Nevertheless, these systems are exposed to some shilling attacks which negatively affect the quality of recommendations. The existing literature on shilling attack detection is extensive and focuses particularly on offline methods. In this paper, a novel online shilling attack detection is proposed to cope with mentioned attacks. This proposed model contributes to improving recommendation quality with immediate interventions to the system. Quality control tools have been used to detect deviations from means for production processes. In the last two decades, these tools have been used for detecting cyber-attacks in computer servers. Therefore two widely used quality tools, Cumulative Sum (CUSUM) and Exponentially Weighted Moving Average (EWMA) are proposed for online shilling attack detection. The Amazon.com dataset is used to show the validity of the proposed model. As a result, a novel online shilling attack detection method is proposed using well-known quality tools and results are presented.**

Keywords— recommender systems, online detection, shilling attacks, EWMA, CUSUM

I. INTRODUCTION

The development of the Internet has brought different developments as well. Online shopping, Internet-based movie watching and music listening sites have also entered our lives. These sites, most of which are established for commercial purposes, use different algorithms to market products that will attract the attention of users. Recommender Systems (RS) are also one of the methods used by these sites.

In its most general definition, recommender systems can be defined as methods that can offer personalized suggestions. With the developing online systems and the tendency of users to shop online, the demand for RS has also increased. The methods used to highlight popular products in the past can now offer special recommenders for each user. RS applies some filters while presenting recommendations. If they are to be classified according to the type of data they use, they can be divided into three classes as user-based, product-based and hybrid methods.

User-based methods are more common in the literature. These methods, which we can also call Collaborative Filtering (CF), basically offer suggestions to users based on the similarities between users. A suggestion is presented by taking into account the users who have made similar scores in the CF methods.

Due to the wide range of application areas, CF methods, which have so many types, the recommendation accuracy rate decreases due to the fake or malicious users added to the systems. Therefore, it is possible to come across many attack detection methods in the literature. Attacks on systems affect the accuracy rate of the method by increasing or decreasing the popularity of the target product.

There are two ways to reduce the effects of these attacks on the system. The first of these is to propose robust methods against attacks on systems; the second is to detect attacks and remove them from the system before recommendations are made. Most of the methods are offline methods that try to remove the attacks from the system before the recommendation process takes place.

In this study, the detection methods that available in the literature were examined and it was concluded that the studies on the detection of instant attacks were open to development. It has been seen that online attack detection systems are important especially in attacks on servers and it is thought that these applications can also be applied to recommendation systems. In this way, it will be possible to monitor the attacks on the systems simultaneously and there will be no need for extra effort for attack detection before recommendations.

Many methods have been proposed for the detection of attacks on servers. These methods include quality control charts and detection methods. Quality control charts control the mean of the data and give a signal in case of a change in the mean of the data. Among the quality control charts frequently used in the literature are Cumulative Sum (CUSUM) and Exponentially Weighted Moving Averages (EWMA) quality control charts. The simplicity of the methods and their applicability for different data types have been seen in existing studies and have been applied to RS for attack detection.

In summary, quality control charts, which are used to detect shifts on the mean, are used for attack detection in RS. Successful results were obtained in the datasets of RS with high autocorrelation and ordinal data type. In line with the results obtained, the usability of quality control charts has been demonstrated for the use of robust RS.

In the following parts of the study, a literature review for attacks on RS, existing attack detection methods and online attack detection systems is presented in the second part. In the third part, the data set used in the study, the structure of the data and the use of quality control charts for attack detection are explained. In the last part, the results and what will be done in future studies are mentioned.

II. ATTACK DETECTION

In its broadest definition, RS can be summarized as recommending the right products to users who do not have any preference information about the products to be recommended by using the similarities of the users or the products. Collaborative Filtering (CF) is a recommendation system model that is frequently used in the literature and offers recommendations for the preferences of similar users in the step of creating recommendations. In other words, it is based on the evaluations of existing users in the dataset. As with all systems, high accuracy rates are aimed at CF recommendation systems. In other words, it is important that the product recommended to the user should be liked by the user. Criteria such as the consistency of the model established on the accuracy rate and the presence of sufficient observations in the data set are effective. In addition to these, one of the most important criteria is attacks on the system. An attack profile defined by Bhaumik et al. is given in Fig. 1 [1]. In general, an attack profile consists of 4 parts. I_S is the set of products that are given scores in which the attack determines the characteristic of the attack with a scoring function. I_F, on the other hand, are the scores made out of the target product to make it difficult to detect the attack. The ratings made here are based on the average and are treated like a real user. In recommendation systems, it is not possible for a user to rate all products. Therefore, I_O is the product set that is not scored. I_T is the product to be attacked. The aim here is to increase or decrease the score of the target product.

I_S	I_F	I_O	I_T
Selected Items	Filler Items	Unrated Items	Target Item

Fig. 1. An Attack Profile Example

A. Shilling Attack Detection for Recommender Systems

Attack detection studies in collaborative filter based recommendation systems focus on shilling attacks, detection of these attacks and collaborative filtering. Various collaborative filtering recommendation systems have been proven to protect against shilling attacks [2]. In order to reduce the effect of shilling attacks, these attacks can be detected and removed from the system before the recommendation system algorithm is created. Another alternative is the creation of robust algorithms against the attacks. Basically, attack detection methods can be divided into supervised, unsupervised and semi-supervised methods. Finally, it is possible to come across studies that are not included in these groups.

The methods that have the class information, in other words, the information of the attack profiles, are called supervised methods. The main methods that we encounter as classification methods are the k nearest neighbors (kNN, C4.5 and support vector machines (SVM). Among the classification methods, SVM has high classification accuracy. Zhang and Zhou proposed an online attack detection method in their study [3]. In the study, first of all, attack profiles were determined and then classification was made according to the detections that made with SVM.

These methods are unsupervised if the class information is not available at the time of grouping the dataset. The methods, also called clustering approach, were first used on recommendation systems by O'Mahony et al. [4]. In the study, the clustering method, which has different applications, was applied to the recommendation systems datasets to increase robustness and to detect fake users. Again, the k-means method, which is very popular among clustering algorithms, has been used for attack detection [5]. The methods in which clustering methods are used by hybridizing with different methods have found a place in the literature [6][7].

With the intensive use of recommendation systems, studies on the detection of attacks on systems have also increased. In this direction, it is possible to come across studies in which supervised and unsupervised methods are used together. Wu et al. using the Naive-based clustering method to determine the attack profiles, then they detected the attack profiles with a classification algorithm with the labels they obtained [8]. Furthermore, the method used was developed and an attack detection method was proposed by using the information on the products [9].

Classification and clustering methods for attack detection methods give very successful results. The methods basically aim to detect users who do not belong to the normal user class or set. Here, users are evaluated as a point in an n-dimensional space, and a classification or clustering is made considering distances. In addition to these, it is possible to come across probability-based methods. Zhang et al. tries to detect fake users in the system using a probability approach [10]. Bhaumik et al. tries to detect attack by using statistical abnormality detection method [1]. There is a normal distribution assumption in their methods. By using X-bar and confidence intervals control limits, values exceed the standard deviation limits were determined.

B. Online Attack Detection

Anomalies can be seen in every system. The well-known anomaly is cyber-attacks in computer networks or servers. These anomalies can be detected in quality control charts which are tools that enable production and service processes to be monitored and controlled using statistical methods through measurable characteristics. The well-known quality control charts are Shewhart [11], CUSUM [12], and EWMA [13] that give a signal if process characteristics of the system change.

Aforementioned charts used different types of attacks in the literature. Cyber-attacks cause serious damages on computer and network systems [14]. Denial of Service (DOS) and Distributed Denial of Service (DDOS) attacks especially effect accessibility of network systems. Shewhart and EWMA charts were used for detection of mentioned attacks and high detection rates were obtained. There are examples in the literature in which CUSUM, EWMA and Shewhart graphics are used to detect the mentioned attacks [15]. In the study, which graph has better performance according to attack intensities is given comparatively. In a similar study, distributed denial of service attacks were applied in real time on time series data with EWMA graph [16]. Another study on this type of attacks was presented by Machaka [17].

There are malicious users trying to break into network systems for different purposes. The study focused on the detection of non-system interventions to source port data [18]. The data was analyzed using Shannon entropy and then the detection was completed with EWMA. In the study [19], a different approach is presented using Renyi entropy. Similar to the previous study, the EWMA graph was used in the detection phase.

With their wide application areas, wireless sensor networks have become promising network solutions. However, the mentioned wireless technology suffers from congestion attacks [20]. In the study, it was aimed to detect abnormal congestions by using the EWMA graph. In line with the given studies, EWMA graphs were used together with fuzzy logic for anomaly detection. In another study, CUSUM and EWMA graphs were used to detect forced entry [21].

A literature on the types of attacks, existing attack detection methods and quality control charts is presented above. From this, we can deduce that the attacks in the recommendation systems are the efforts to change the average score of the attacked product in the normal process. On the other hand, if the quality control charts are defined as systems that warn the system analyst in cases where the process average is deteriorated, it is expected that the attacks on the recommendation systems will be detected online. In the following section, the application of quality control charts and their application to datasets will be presented.

III. MATERIALS AND METHODS

CF methods require historical data to be able to present recommendations. The similarities on the existing user ratings are evaluated and recommendations are offered to the users. Data stored in the form of time series is needed for the online attack detection system to be used in the study. Datasets stored by Amazon.com were used in this study. General information about the data structure will be given in detail in the next section.

A. Data

Amazon datasets consist of time series data that stores 82.6 million ratings on 10 million products by approximately 21 million users. Data consist of four information as, user ID, item ID, score, and timestamp. User ID and item ID contain alpha-numeric values so data pre-processing is applied to simplicity. Alpha-numeric values are converted numeric values.

B. CUSUM

In the CUSUM method, the deviations from the mean value are collected cumulatively and marked on the graph.

$$C_i = \sum_{i=1}^{i}(x_i - \mu_0) \quad (1)$$

In Equation 1, x_i and $\mu_0$indicates i^{th}score and mean respectively. C_i indicates the cumulative sum, including i^{th}score. C_i values exhibit a zero-sum distribution when the process is under control. If the mean of the process $\mu_1$is greater than the target value, a positive trend is seen in the C_i values. Conversely, if $\mu_1$is less than the target value (μ_0), a negative trend is seen in C_i values. In CUSUM graphs, 2 statistics are kept as increases for C_i^+ and decreases for C_i^- and calculated as in Equation 2 and 3 [22].

$$C_i^+ = max[0, x_i - (\mu_0 + K) + C_{i-1}^-] \quad (2)$$

$$C_i^- = max[0, (\mu_0 - K) - x_i + C_{i-1}^-] \quad (3)$$

The initial values $C_i^-$and C_i^+ are zero. K is in the range of the mean value to be determined and the target value and calculated as in Equation 4, 5, and 6.

$$\mu_1 = \mu_0 + \delta\sigma \quad (4)$$

$$\delta = \frac{|\mu_1 - \mu_0|}{\sigma} \quad (5)$$

$$K = \frac{\delta}{2}\sigma = \frac{|\mu_1 - \mu_0|}{2} \quad (6)$$

The process is considered out of control if one of the $C_i^-$and C_i^+ values exceeds H which is usually chosen as 5 times the standard deviation

C. EWMA

EWMA charts were first proposed by Roberts in 1959 as an alternative method to CUSUM quality charts [13]. While calculating the current statistic, the contribution of the past data to this statistic decreases exponentially. It has been proposed to detect small deviations from the mean relative to Shewhart charts as in CUSUM charts. The EWMA statistic is defined as in Equation 7.

$$Z_i = \lambda x_i + (1 - \lambda)Z_{i-1} \quad (7)$$

Here x_i, Z_i, and λ represent i^{th}score, new moving average and weighting parameter, respectively The weighting parameter takes values in the range of $0 < \lambda \leq 1$. As the λ value increases, the effect of the past scores on the current statistics will increase. Upper and lower control limits for EWMA charts are given in Equation 8 and 9, respectively.

$$\mu_0 + L\sigma\sqrt{\frac{\lambda}{2-\lambda}[1 - (1-\lambda)^{2i}]} \quad (8)$$

$$\mu_0 - L\sigma\sqrt{\frac{\lambda}{2-\lambda}[1 - (1-\lambda)^{2i}]} \quad (9)$$

D. Experiment Evaluation

Statistical quality control is implemented in two phases. In Phase I, a part of data is used to determine the upper and lower control limits. In addition to, descriptive statistics are calculated in this phase, like as mean value and standard deviation of the data. Then, EWMA and CUSUM statistics are calculated, and given scores locate between the mentioned limits. A numerical example is given Fig.2 for EWMA and Fig.3 for CUSUM. According to mentioned figures, detected points are same for both methods. In phase I, points that exceed the limits, removed from data and limits are calculated again.

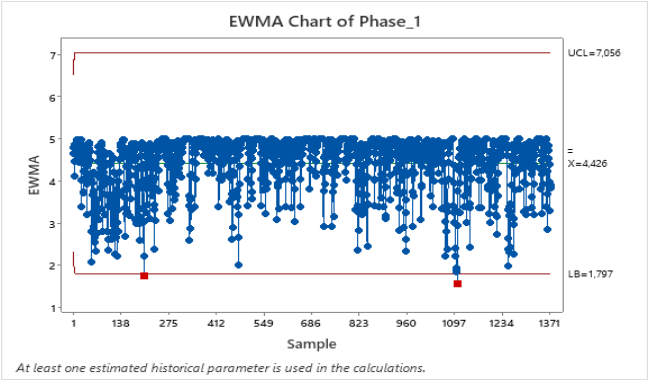


Fig. 2. Phase 1 Chart of EWMA

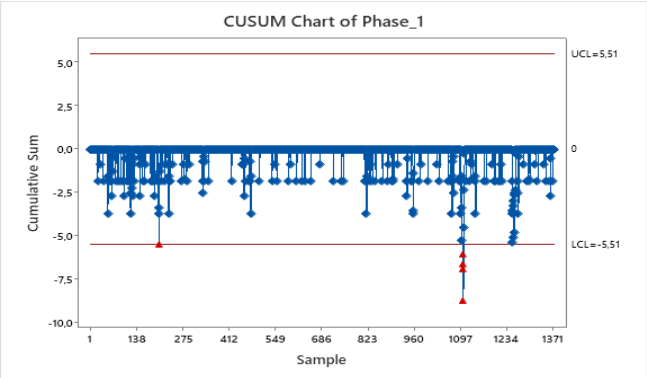


Fig. 3. CUSUM Chart of Phase 1

In Phase II, new data located to obtained charts to test of model consistency. EWMA and CUSUM charts are given for phase 2 in Fig. 4 and Fig. 5, respectively. These figures show that limits that found in phase 1 is consistent and these charts can be used to further monitoring process.

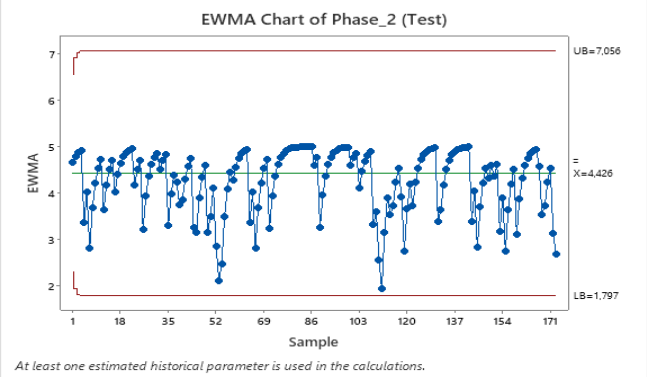


Fig. 4. Phase 2 Chart for Monitoring (EWMA)

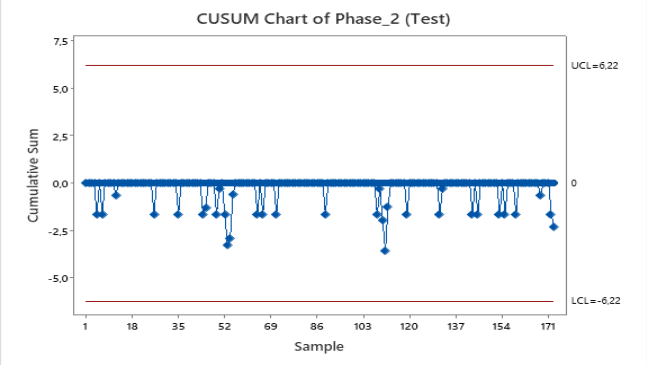


Fig. 5. Phase 2 Chart for Monitoring (CUSUM)

EWMA and CUSUM charts detect the out-of-control points as shown above figures. In Fig. 6 and Fig. 7 some attacks are added to data. These attack are 20 to 24, 62 to 65, and 103 to 105 points as can be shown below figures. Both methods give a signal on attack points.

A numerical example is given in this section. Quality control charts are applied to scores that given by user in recommender systems. Phase I and Phase II is applied mentioned data and finally attack detection process is applied. Both methods are detect the attacks successfully. Results are show that both methods can be used online and simultaneously attack detection for recommender systems.

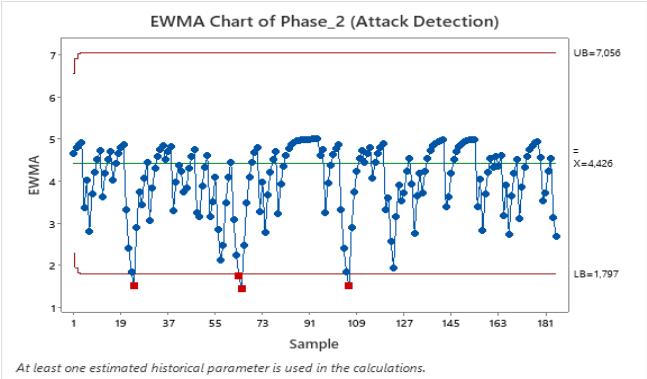


Fig. 6. Phase 2 Chart for Attack Detection (EWMA)

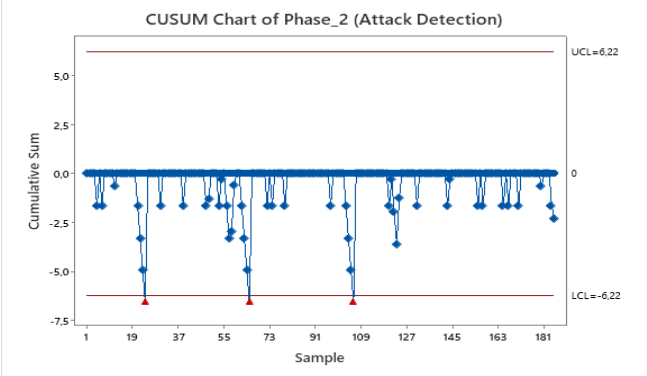


Fig. 7. Chart for Attack Detection (CUSUM)

IV. Conclusions and Future Directions

With the expanding usage area of the Internet, many platforms are moving to the Internet environment. The leading of these platforms are the shopping, movie and music sectors. RS are needed for user satisfaction on commercial platforms. However, RS are subject to many attacks of different types. Fake or malicious users can seriously damage the recommendation mechanism of the RS methods, hinder the accuracy of the methods and offer false recommendations to users.

In this study, we focused on the detection of attacks in order to propose methods against attacks. The applicability of quality control charts, which produce effective results in attacks on servers, on recommendation systems was discussed, and applications on Amazon.com datasets were presented. The attacks on the RS were detected by the CUSUM and EWMA charts, which produced effective results in tracking the shifts in the mean of the time series data. Studies have shown that attacks on recommendation systems, including small deviations, can be detected with quality control charts. An application was presented on a small data set in the study. Attack detection was provided with EWMA and CUSUM charts. For future research, best parameters for these charts can be discussed.

Acknowledgment

This study is supported by Eskisehir Technical University, Scientific Research Projects Committee (ESTUBAP- 20DRP026).

References

[1] R. Bhaumik, C. Williams, B. Mobasher, and R. Burke, “Securing collaborative filtering against malicious attacks through anomaly

detection," *AAAI Work. - Tech. Rep.*, vol. WS-06-10, no. Shewart, pp. 50–59, 2006.
[2] A. Bilge, I. Gunes, and H. Polat, "Robustness analysis of privacy-preserving model-based recommendation schemes," *Expert Syst. Appl.*, vol. 41, no. 8, pp. 3671–3681, 2014,
[3] F. Zhang and Q. Zhou, "HHT-SVM: An online method for detecting profile injection attacks in collaborative recommender systems," *Knowledge-Based Syst.*, 2014,
[4] M. P. O'Mahony, N. J. Hurley, and G. C. M. Silvestre, "Collaborative filtering – Safe and sound?," 2003.
[5] R. Bhaumik, B. Mobasher, and R. Burke, "A Clustering Approach to Unsupervised Attack Detection in Collaborative Recommender Systems," *Proc. 7th IEEE Int. Conf. data Min.*, pp. 181–187, 2011.
[6] Z. Cheng and N. Hurley, "Robustness analysis of model-based collaborative filtering systems," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6206 LNAI, no. May, pp. 3–15, 2010,
[7] B. Mehta and W. Nejdl, *Unsupervised strategies for shilling detection and robust collaborative filtering*, vol. 19, no. 1-2 SPEC. ISS. 2009.
[8] Z. Wu, J. Wu, J. Cao, and D. Tao, "HySAD: A semi-supervised hybrid shilling attack detector for trustworthy product recommendation," 2012.
[9] Z. Wu, Y. Wang, Y. Wang, J. Wu, J. Cao, and L. Zhang, "Spammers detection from product reviews: A hybrid model," 2016.
[10] S. Zhang, Y. Ouyang, J. Ford, and F. Makedon, "Analysis of a low-dimensional linear model under recommendation attacks," *Proc. Twenty-Ninth Annu. Int. ACM SIGIR Conf. Res. Dev. Inf. Retr.*, vol. 2006, no. January, pp. 517–524, 2006,
[11] W. A. Shewhart, "Quality Control Charts," *Bell Syst. Tech. J.*, vol. 5, no. 4, pp. 593–603, 1926,
[12] E. S. Page, "Continuous inspection schemes," *Oxford Univ. Press behalf Biometrika Trust*, vol. 1, no. 2, pp. 100–115, 1954.
[13] S. W. Roberts, "Control Chart Tests Based on Geometric Moving Averages," *Technometrics*, vol. 1, no. 3, pp. 239–250, 1959,
[14] F. Harrou, B. Bouyeddou, Y. Sun, and B. Kadri, "Detecting cyber-attacks using a CRPS-based monitoring approach," *Proc. 2018 IEEE Symp. Ser. Comput. Intell. SSCI 2018*, pp. 618–622, 2019,
[15] B. Bouyeddou, F. Harrou, Y. Sun, and B. Kadri, "Detecting SYN flood attacks via statistical monitoring charts: A comparative study," *2017 5th Int. Conf. Electr. Eng. - Boumerdes, ICEE-B 2017*, vol. 2017-Janua, pp. 1–6, 2017,
[16] X. Yue, X. Mo, C. Wang, and X. Yao, "Research on real-time flow abnormal traffic detection system based on DDoS attack," *Adv. Intell. Syst. Comput.*, vol. 541, pp. 206–212, 2017,
[17] P. Machaka, A. Bagula, and F. Nelwamondo, "Using exponentially weighted moving average algorithm to defend against DDoS attacks," *2016 Pattern Recognit. Assoc. South Africa Robot. Mechatronics Int. Conf. PRASA-RobMech 2016*, 2017,
[18] S. Ransewa, N. Elz, N. Thanon, and S. Intajag, "Anomaly detection using source port data with shannon entropy and EWMA control chart," *Int. Conf. Control. Autom. Syst.*, vol. 2018-Octob, no. Iccas, pp. 596–601, 2018.
[19] R. Yan, "Combining Renyi Entropy and EWMA to Detect Common Attacks in Network," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 30, no. 10, pp. 1–23, 2016,
[20] O. Osanaiye, A. S. Alfa, and G. P. Hancke, "A statistical approach to detect jamming attacks in wireless sensor networks," *Sensors (Switzerland)*, vol. 18, no. 6, 2018,
[21] D. Sklavounos, A. Edoh, and M. Plytas, "A Statistical Approach Based on EWMA and CUSUM Control Charts for R2L Intrusion Detection," *Proc. - 2017 Cybersecurity Cyberforensics Conf. CCC 2017*, vol. 2018-Septe, pp. 25–30, 2017,
[22] D. C. Montgomery, *Introduction to Statistical Quality Control, Sixth Edition*. 2009.

Infrastructure Building Platform for Enterprise Private Cloud Systems: An Apllication with Machine Learning

Halil ARSLAN
Computer Engineering
Sivas Cumhuriyet University
Sivas, Turkey
harslan@cumhuriyet.edu.tr

Yasin GÖRMEZ
Management Information Systems
Sivas Cumhuriyet University
Sivas, Turkey
yasingormez@cumhuriyet.edu.tr

İbrahim Ethem DADAŞ
BCA
Detay Danışmanlık
İstanbul, Turkey
ibrahim.dadas@detaysoft.com

Erdi ŞEVRAN
SAPABHR
Detay Danışmanlık
İstanbul, Turkey
erdi.sevran@detaysoft.com

Veysel GUNDUZ
BCC
Detay Danışmanlık
İstanbul, Turkey
veysel.gunduz@detaysoft.com

***Abstract*— Platform independent operation and horizontal scalability for applications have become more important than before with the increase in the number of users. Although these processes can be done easily with the help of systems such as Docker and Kubernetes the installation and configuration stages of these technologies can take a lot of time to learn and implement. Nowadays, many complex problem can be solved with the machine learning methods. This machine learning model have to be suitable with the current systems of the companies. In this study, a user-friendly cloud system has been developed that simplifies the installation stages of container and container orchestration tools that are one of the most frequently used technologies of recent times. In the later stages of the study, a system to predict are of interest using Random Forest is proposed. This system was generated using our cloud platform and showed that, a complex cluster system can be developed using our system without any DevOps knowledge.**

Keywords—Cloud Computing, DevOps, Machine Learning, Platform Independency, Scaling

I. INTRODUCTION

Machine learning (ML) algorithms are used to solve many problems from several fields such as sentiment analysis, disease prediction, localization systems and smart cities [1]–[4]. ML algorithms are also used in industry because of the good results were obtained with them [5], [6]. Although, sufficient accuracy results are enough to use ML algorithms, some other requirements are necessary to integrate them into applications. Industry applications may run on several operating system and several version of same operating system. Different requirements may be needed to run application on different operating system or version. Considering the increase in the number of operating systems platform independency have become more important than before.

Large companies have several applications that developed using different programming languages. These applications must work in an integrated way in order for the business processes to run smoothly. Solutions of this problem is, developing the system using micro-service architecture. Considering the number of users and potential increasing of it, applications have to be scaled horizontally.

The other problem of today's applications are data transferring. As the number of data transferred per second increases, data loss may occur. Advance message queuing algorithms are used to prevent the data loss problem. Trained model have to be published to make machine learning algorithms useful for applications. Advance message queuing algorithms have to be used in this kind of applications to respond to all request that came to machine learning model.

Nowadays, container and container orchestration tools are commonly used to make applications platform independent and horizontally scaled. Thanks to some open source tools such as Docker and Kubernetes, platform independency and horizontal scaling can be done easily. However, the installation and configuration stages of these technologies can take a lot of time to learn and apply. This unnecessary effort is a burden on companies. In this study, a user-friendly system has been developed that simplifies the installation stages of containers and container orchestration tools. Docker and Kubernetes can be installed and configured easily by using this system. In the ongoing stages of the study, a cloud system was generated using this system and it was integrated with the trained ML model that taken from our previous work [7]. An intermediate layer that contains Kafka as a message broker was added to system to prevent data loss to ML model.

II. LITERATURE REVIEW

To date, many cloud system have been studied commonly in the literature. Mariani et al. proposed a machine learning models to localize faults in cloud system [8]. Ferry et al. discuss the techniques and methods for deployment, monitoring, and adaptation concerns of multi-cloud systems at design-time and their enactment at run-time [9]. Kumar et al. proposed discrete binary cat swarm optimization to optimize workflow schedule length for cloud systems [10]. Stephens showed the effect of cloud systems on the earth atmosphere and climate problem in the coming decade [11]. Babaoglu et al. proposed fully decentralized peer-to-peer cloud system that allows organizations or even individual to build a computing infrastructure out of existing resources, which can be easily allocated among different tasks [12]. In addition to these research, cloud systems also used in to machine learning applications. Islam and Manivannan

proposed a long short-term memory networks (LSTM) to predict task failures and they obtained 87% accuracy [13]. Kirchoff et al. compared ARIMA, multi-layer perceptron (MLP) and gated recurrent unit (GRU) for workload prediction techniques to help administrators to choose the most appropriate algorithm for their specific applications [14]. Motwani et al. developed machine learning models to predict credit card payment defaulter that runs on Microsoft Azure Machine Learning platform [15]. Suresh and Ganesh Kumar developed a decision tree based algorithm to select suitable prediction model for user need to maximize profit of both user and cloud machine learning service providers [16]. Abdelaziz et al. proposed particle swarm optimization model to select optimum virtual machine for health care services system [17]. Catak and Balaban trained Support Vector Machines (SVM) on cloud system by using map reduce to train large dataset in less time [18].

III. Proposed System

In this study a cloud system was developed to install and configure kubernetes and Docker without writing any code or script. In first step of the study, the business processes of the companies that our company is affiliated with were examined and the technologies that are frequently used or needed in current applications were determined by taking ideas from the companies. As a result of research, install and configuration scripts for Docker, docker-swarm, kubernetes, Podman, Vagrant and Virctualbox were developed and added to system. Thanks to this scripts any user can generate a cluster using these technologies automatically. No code is required for any user to generate special cluster. After that, a user-friendly interface has been developed and added to the system to make in more useful for any user that do not have any information about DevOps process. Figure 1 shows the screenshot of the developed interface.

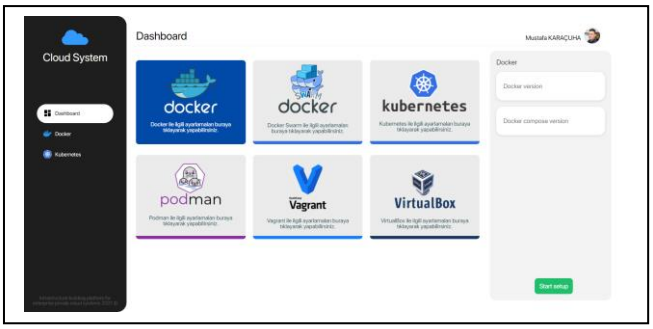


Fig. 1. Screenshot of the cloud platform main page

Fig. 2. User Interface of the cloud platform Docker container

Docker is a container technologies that enables user to virtualize their application into small packages. Technologies should be installed into Docker container to make it useful. In the next step, script to install databases, which are MySQL, MongoDB, PorteSQL and Cassandra, were added to platform to let user install and configure them automatically. User interface of the Docker system shown in Figure 2.

In the third step of the study, the trained ML model from our previous work is taken to design decision support system [7]. In this model, the individuals' areas of interest were estimated through machine learning methods in order to use it as an infrastructure for personalized foreign language learning (the foreign language learning platform reachable at https://listening.detayteknoloji.com). This model trained using 164 samples where each sample consist of 11 features. Six area of interests, which are technology, health, business life, different cultures, sports and fine arts, are predicted with multi-label approach using Random Forest algorithm. The accuracy of the model was 78.12%. Our purpose is to integrate ML to cloud system. For this purpose, a micro service was developed using Python language. In this service the trained ML model are loaded and published via HTTP protocol. As mentioned before, the number of request may be too much in the real time systems. To prevent data loss problem a Kafka, which run on Docker container, installed using our platform. The architecture of whole system shown in Figure 3.

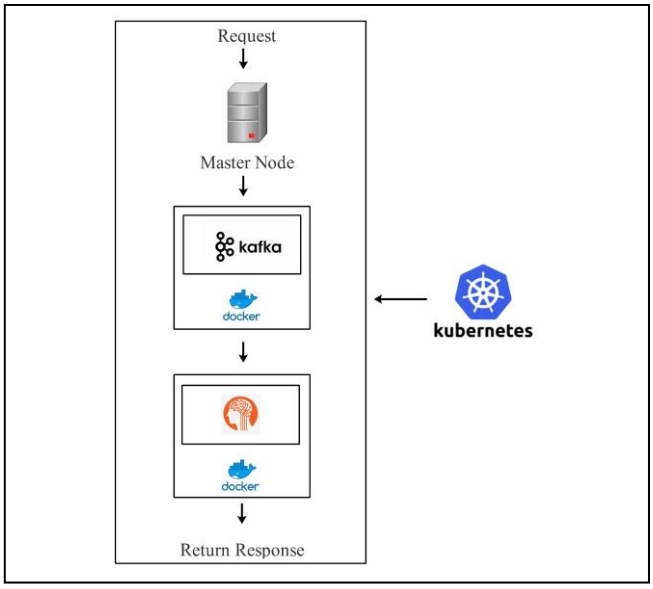


Fig. 3. Architecture of whole cluster system with an application using machine learning.

In this platform request are first caught by master node. The purpose of master node is controlling the workers load and working status. The master node forward the request to optimal worker. The workers contains the Kafka broker and ML model. The number of master node and works can be increased according to the number of requests. Each worker run on Docker container and whole system manage by kubernetes. This system generated using our cloud platform.

IV. Conclusions

As a result of the researches, it has been understood that there is no Platform as a Service (PaaS) solution for container-based platforms. Technologies such as Kubernetes and Docker the tools required for these technologies are manually installed in existing systems. The product to be obtained as a result of the study will be offered to cloud-based companies

as a PaaS solution. It will increase efficiency in internal processes to ease the hassle of installing Kubernetes and Docker platforms and optimize installation configurations. Thanks to this cloud system technologies and tools for the Docker and kubernetes will be automatically installed as a package with the system to be developed. In the further steps of the study, the cluster is generated using our cloud platform for a machine learning applications that predicts the area of interests. This model, which was taken from our previous work [7], is the machine learning model of learning system platform. As a result we showed that complex cluster can be setup and installed using our cloud platform easily without any knowledge of DevOps process. As a future work, we will setup more complex cluster that contains several technologies such as ElasticSearch, NoSql database and Redis. Intercalary, we will make a load test to system and we will analyses the performance of cluster that generated using our platform.

ACKNOWLEDGMENT

This study is an output of studies conducted in DetaySoft research and development center. We appreciate their support

REFERENCES

[1] S. Khan, M. Hussain, H. Aboalsamh, and G. Bebis, "A comparison of different Gabor feature extraction approaches for mass classification in mammography," Multimed. Tools Appl., vol. 76, no. 1, pp. 33–57, Jan. 2017, doi: 10.1007/s11042-015-3017-3.

[2] Ankit and N. Saleena, "An Ensemble Classification System for Twitter Sentiment Analysis," Procedia Comput. Sci., vol. 132, pp. 937–946, Jan. 2018, doi: 10.1016/j.procs.2018.05.109.

[3] J. Röbesaat, P. Zhang, M. Abdelaal, and O. Theel, "An Improved BLE Indoor Localization with Kalman-Based Fusion: An Experimental Study," Sensors, vol. 17, no. 5, Art. no. 5, May 2017, doi: 10.3390/s17050951.

[4] A. Koesdwiady, R. Soua, and F. Karray, "Improving Traffic Flow Prediction With Weather Information in Connected Cars: A Deep Learning Approach," IEEE Trans. Veh. Technol., vol. 65, no. 12, pp. 9508–9517, Dec. 2016, doi: 10.1109/TVT.2016.2585575.

[5] Z. Ge, Z. Song, S. X. Ding, and B. Huang, "Data Mining and Analytics in the Process Industry: The Role of Machine Learning," IEEE Access, vol. 5, pp. 20590–20616, 2017, doi: 10.1109/ACCESS.2017.2756872.

[6] I. S. Candanedo, E. H. Nieves, S. R. González, M. T. S. Martín, and A. G. Briones, "Machine Learning Predictive Model for Industry 4.0," in Knowledge Management in Organizations, Cham, 2018, pp. 501–510. doi: 10.1007/978-3-319-95204-8_42.

[7] Details are hidden under the double-blind privacy policy.

[8] L. Mariani, C. Monni, M. Pezzé, O. Riganelli, and R. Xin, "Localizing Faults in Cloud Systems," in 2018 IEEE 11th International Conference on Software Testing, Verification and Validation (ICST), Apr. 2018, pp. 262–273. doi: 10.1109/ICST.2018.00034.

[9] N. Ferry, A. Rossini, F. Chauvel, B. Morin, and A. Solberg, "Towards Model-Driven Provisioning, Deployment, Monitoring, and Adaptation of Multi-cloud Systems," in 2013 IEEE Sixth International Conference on Cloud Computing, Jun. 2013, pp. 887–894. doi: 10.1109/CLOUD.2013.133.

[10] B. Kumar, M. Kalra, and P. Singh, "Discrete binary cat swarm optimization for scheduling workflow applications in cloud systems," in 2017 3rd International Conference on Computational Intelligence Communication Technology (CICT), Feb. 2017, pp. 1–6. doi: 10.1109/CIACT.2017.7977296.

[11] G. L. Stephens, "Cloud Feedbacks in the Climate System: A Critical Review," J. Clim., vol. 18, no. 2, pp. 237–273, Jan. 2005, doi: 10.1175/JCLI-3243.1.

[12] O. Babaoglu, M. Marzolla, and M. Tamburini, "Design and implementation of a P2P Cloud system," in Proceedings of the 27th Annual ACM Symposium on Applied Computing, New York, NY, USA, Mar. 2012, pp. 412–417. doi: 10.1145/2245276.2245357.

[13] T. Islam and D. Manivannan, "Predicting Application Failure in Cloud: A Machine Learning Approach," in 2017 IEEE International Conference on Cognitive Computing (ICCC), Jun. 2017, pp. 24–31. doi: 10.1109/IEEE.ICCC.2017.11.

[14] D. F. Kirchoff, M. Xavier, J. Mastella, and C. A. F De Rose, "A Preliminary Study of Machine Learning Workload Prediction Techniques for Cloud Applications," in 2019 27th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), Feb. 2019, pp. 222–227. doi: 10.1109/EMPDP.2019.8671604.

[15] A. Motwani, P. Chaurasiya, and G. Bajaj, "Predicting Credit Worthiness of Bank Customer with Machine Learning Over Cloud," Int. J. Comput. Sci. Eng., vol. 6, pp. 1471–1477, Jul. 2018, doi: 10.26438/ijcse/v6i7.14711477.

[16] A. Suresh and P. Ganesh Kumar, "Optimization of Metascheduler for Cloud Machine Learning Services," Wirel. Pers. Commun., vol. 114, no. 1, pp. 367–388, Sep. 2020, doi: 10.1007/s11277-020-07367-2.

[17] A. Abdelaziz, M. Elhoseny, A. S. Salama, and A. M. Riad, "A machine learning model for improving healthcare services on cloud computing environment," Measurement, vol. 119, pp. 117–128, Apr. 2018, doi: 10.1016/j.measurement.2018.01.022.

[18] F. O. Catak and M. E. Balaban, "CloudSVM: Training an SVM Classifier in Cloud Computing Systems," in Pervasive Computing and the Networked World, Berlin, Heidelberg, 2013, pp. 57–68. doi: 10.1007/978-3-642-37015-1_6.

Machine Learning based Radar Target Detection Application

Alparslan FİŞNE
Aselsan Research Center
ASELSAN
Ankara, Türkiye
afsine@aselsan.com.tr

Faruk YAVUZ
Aselsan Research Center
ASELSAN
Ankara, Türkiye
farukyavuz @aselsan.com.tr

***Abstract*—This study proposes a real-time realization for a machine learning (ML) based radar target detector using central processing units (CPU) and graphics processing units (GPU). The ML based radar target detector takes the range-Doppler ambiguity function as its input, and outputs the range and Doppler information of a single target through employing Convolutional Neural Networks (CNN). Radar systems typically have hard constraints on the execution times of detection algorithms. In order to meet these timing requirements, the CNN-based ML detector needs a speedup through the use of parallel programming architectures as opposed to serial CPU programming. The parallel GPU processing architecture proposed in this work achieves 41x speed improvement compared to serial CPU implementation. This speed up allows the ML radar detector to successfully comply with real-time requirements of radar systems.**

Keywords—Machine learning, radar target detection, convolutional neural networks, parallel programming, GPGPU.

I. INTRODUCTION

Since the beginning of 20th century, radar has been widely used for military and civilian applications [1]. A pulse-Doppler radar provides the range and velocity information of targets by digitizing the received echoes reflected from these objects, and registering the time and frequency information into a data matrix. This data matrix represents the range and Doppler informationalong its time and frequency axes, respectively. Conventional pulse-Doppler radars carry out target detection by comparing the aforementioned range-Doppler data matrix with a certain threshold obtained through Constant False Alarm Rate (CFAR) techniques [2].

Machine Learning (ML) techniques learn representative features of a domain through large datasets by scanning different abstraction layers. Speech recognition, object recognition, natural language processing are some of the fields where ML techniques achieve considerable improvements compared to traditional algorithms [3]. Radar signal processing is another avenue where ML techniques could potentially offer superior performance [4]. Such techniques may learn additional features to help refine amplitude-based thresholds for target detection, and improve detection performance of traditional radar systems.

This work focuses on such a ML-based radar target detector and its real-time implementation aspects. The radar target detector adopted in this study takes the range-Doppler ambiguity function as its input. The detection algorithm uses Convolutional Neural Networks (CNN), and outputs the range and Doppler information of a single target, if there is any [5]. Similar to the application of "LeNet-5" algorithm for handwritting classificiation [6], introduction of the CNN-based detector to radar target detection problem improves detection accuracy [5]. Due to the nature of pulse-Doppler radars, the analyzed CNN-based detector needs to satisfy certain real-time processing requirements.

In order to meet real-time processing requirements of radar systems, parallel programming architectures are typically employed as opposed to traditional serial CPU processing. It has been stated in the literatüre that concurrent CPU programming is not efficient for two dimensional (2D) and multichannel convolutional operations with large size data [7]. 2D convolutions are more efficient when implemented with the NVIDIA cuDnn library [8]. CNN-based radar target detection is feasible with GPU concurrent programming using many blocks and duties. This work proposes and contributes a real-time implementation for the adopted ML-based radar target detector [5] using GPU concurrent programming techniques, to radar industry.

The rest of the paper is organized as follows. Section II reviews the CNN-based radar target detection algorithm, and the motivation of the study. Section III details the proposed GPU parallel archtitectures. Section IV presents the performance results of the designed CPU/GPU archtitectures, and highlights the speedup achieved by the GPU architecture. Finally, Section V concludes the paper and lays out future work.

II. RADAR TARGET DETECTION WITH MACHINE LEARNING

This section summarizes the CNN-based radar target detection algorithm previously proposed in [5], and builds up the motivation for the proposed study.

A. CNN based Radar Target Detection

Radar technology encompasses several different hardware and software configurations, depending on desired sensor capabilities. The radar studied in this work is a pulse-Doppler radar having a single monostatic omnidirectional antenna. For this radar configuration, the antenna transmits specifically designed pulses and listens to the received echoes from targets. Table I provides basic specifications of the analyzed radar system.

TABLE I. RADAR SPECIFICATIONS

Number of Range Cells	Number of Doppler Cells	Radar Processing Period
9000	20	9 ms

As can be observed from Table 1, the radar system generates 9000 fast-time samples and 20 slow-time samples within a Coherent Processing Interval (CPI). Traditional target detection algorithms proceed by generating the range-Doppler ambiguity function through processing of this data matrix. Each entry of the range-Doppler ambiguity matrix is referred to as a *cell*, which describes the position in the range-Doppler space. The signal amplitudes of reference cells are used to determine the necessary CFAR threshold for a cell under test.

There are a few different types of CFAR algorithms, each specializing on specific target detection scenarios [2]. The target detector adopted in this study takes the range-Doppler ambiguity matrix as an image input, and uses a CNN-based architecture for target detection. The range-Doppler ambiguity matrix is defined over the range and Doppler axes as follows:

$$B[r,d] = \sum_{m=0}^{D-1}\sum_{n=0}^{R-1} x[n,m]x_r^{\star}[n-r,m-d]e^{-\frac{j2\pi md}{D}} \quad (1)$$

In (1), $\boldsymbol{x}$ is the received data matrix, $\boldsymbol{x_r^{\star}}$ is the complex conjugate of the reference waveform, L is the number of range cells, M is the number of pulses in one CPI, n is the range axis, m is the Doppler axis, d/M is the normalized Doppler frequency. The inner summation of (1) can be rewritten as:

$$T[r,d,m] = \sum_{n=0}^{L-1} x[n,m]\, x_r^{*}[n-r,m-d] \quad (2)$$

In (2), $\boldsymbol{T[r,d,m]}$ is the output of the matched filter over a specific pulse period [9]. Equation (2) transforms (1) into the following new form:

$$B[r,d] = \sum_{m'=0}^{M-1} T[r,d,m]\, e^{-\frac{j2\pi md}{D}} \quad (3)$$

As can be seen from (3), the ambiguity function can be calculated by taking the Fourier transform of the matched filter output $\boldsymbol{T[r,d,m]}$ [10]. Consequently, input of the CNN block can be generated by matched-filtering the received signal along the range axis, and then subsequently applying a Fourier transform along the Doppler axis. The resulting ambiguity matrix is then subjected to an absolute square operation for each cell. Figure 1 summarizes the algorithm of the CNN-based radar target detector. Radar target detection is performed on the absolute squared ambiguity matrix.

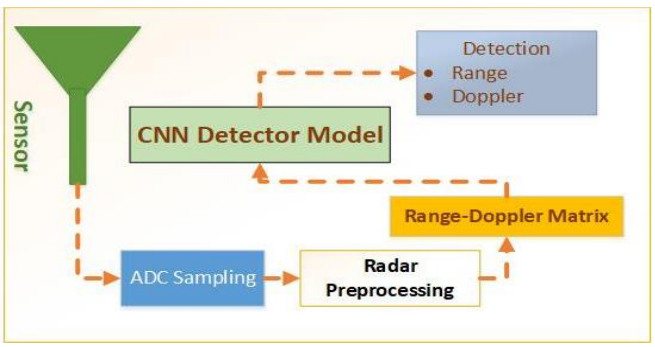


Fig. 1. CNN-based target detection algorithm.

As detailed in [5], the CNN target detector model is simplified to generate 29 different labels in an effort to reduce the computational load of the training procedure. One of the 29 labels corresponds to the scenario where there is no target, whereas the rest represent specific target positions on range-Doppler space. The latter 28 positions are on a grid which is generated by dividing the range and Doppler axes into 4 and 7 sections with 10 km and 30 m/s steps, respectively [5]. Figure 2 shows the structure of the CNN architecture.

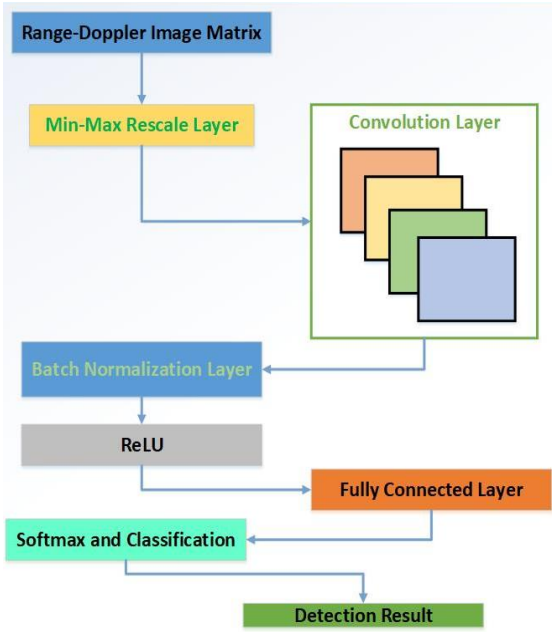


Fig. 2. CNN Architecture

B. *Motivation*

It has been shown in [5] that the CNN-based target detector achieves better detection performance compared to traditional CFAR techniques. To benefit from this performance improvement in a practical radar sensor, this detection algorithm should run in real-time on the radar processor. This practical aspect constitutes the main motivation for this study: Within the scope of this work, we develop computational architectures with parallel programming for CNN-based radar target detection to meet real-time operation requirements.

III. THE PARALLEL PROGRAMMING FOR CNN MODEL

Parallel programming efforts begin with the analysis of serial CPU performance of the CNN model in order to identify computational bottlenecks. Figure 3 shows serial CPU performance profiles for individual CNN layers in terms of their CPU runtime percentages. The multi-channel 2D convolution layer of the CNN model performs convolution filtering on the radar image matrix. This 2D convolution filtering step has a significant impact on the overall processing time because of its high computational complexity. In addition, another computation bottleneck occurs due to the large matrix multiplication operation in the Fully Connected layer according to results of Figure 3. Since the latter operation is performed with serial processing, its computation time is significantly high. Parallel computation of the CNN model is primarily handled in CPU for layers where serial computation causes substantial performance drawbacks.

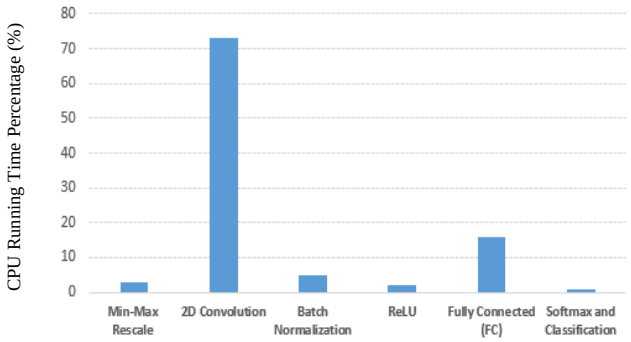


Fig. 3. Serial CPU performance profiles for each CNN layer.

Parallel programming is achieved by performing core optimizations (SIMD) and inter-core optimizations (OpenMP) in the CPU [11-12]. In this work, parallelized 2D convolution and matrix multiplication operations run with higher performance than their serial counterparts through use of Intel MKL [13]. Table II summarizes the parallel processing methods applied for each layer.

TABLE II. CPU PARALLEL PROCESSING METHODS

CNN Model Layers	Optimization Methods
Min-Max Rescale	OpenMP + SIMD
2D Convolution	MKL
Batch Normalization	OpenMP + SIMD
ReLU	OpenMP + SIMD
Fully Connected Layer	MKL

We use the parallel processing based functions of Intel MKL for parallel computation of 2D Convolution and matrix multiplication [13]. In our design, we prefer Intel AVX instruction set for the use of SIMD architecture [14]. However, since the 2D Convolution process includes a step-by-step scanning process and this process has a structure that renders cache optimization challenging, we could not get the desired computational efficiency in the CPU. There are a few studies in the literature which encourage the use of GPUs in machine learning, especially for convolution processes, due to the dramatic reduction of training times [15]. Motivated with this observation, this work further examines the use of GPU for convolution processes. For this purpose, we use the relevant function of NVIDIA cuDNN Library for the 2D Convolution operation [16]. Moreover, we introduce a CPU-GPU heterogeneous computing architecture for parallel processing with GPU. The proposed design runs processes suitable for parallel processing on the GPU, and applies the remaining processes to the CPU. Figure 4 shows the proposed CPU-GPU heterogeneous architecture. Almost all operations run on GPU due to their large inputs, except for Softmax and classification layer operations. A manageable performance loss is expected due to data transfer between CPU and GPU for this heterogeneous architecture.

IV. BENCHMARK TESTS AND RESULTS

For real-time radar target detection, run time of the CNN model should be shorter than the data sampling period. We measure computation times via benchmark tests for our proposed CPU parallel processing and CPU-GPU heterogeneous architectures. Table III lists the computational units used in benchmark tests along with their specifications.

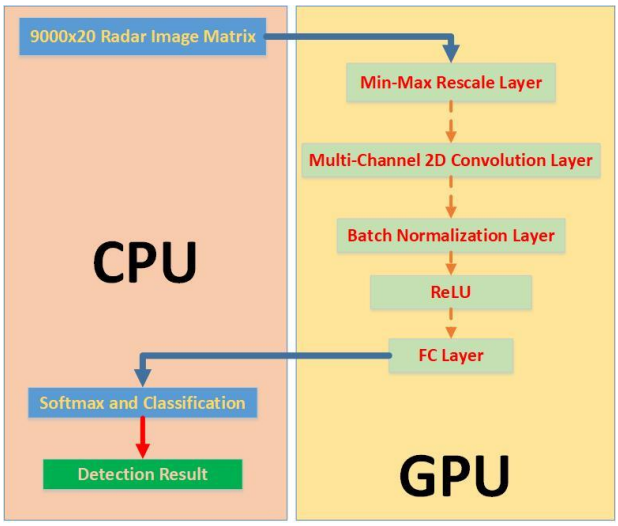


Fig. 4. CPU-GPU heterogeneous architecture

TABLE III. COMPUTATION UNITS FOR BENCHMARK TESTS

Unit	Frequency (GHz)	# of Cores	Price ($)
Intel i9-9900K CPU	3.6	8	435 [17]
Intel Xeon E5-2637 CPU	3.5	4	1100 [18]
NVIDIA GTX 1080 GPU	1.6	2560	350 [19]

Computation times are obtained from an average over 100 measurements. Figure 5 presents measured computation times and calculated speedup factors for each architecture. A quick glance at Figure 5 reveals that the graphics processor offers a remarkable acceleration compared to other computational units. In addition, Intel i9 CPU achieves 1.4x speedup compared to Intel Xeon CPU. Moreover, GPU based computation provides 41.3x speedup compared to Intel i9 CPU.

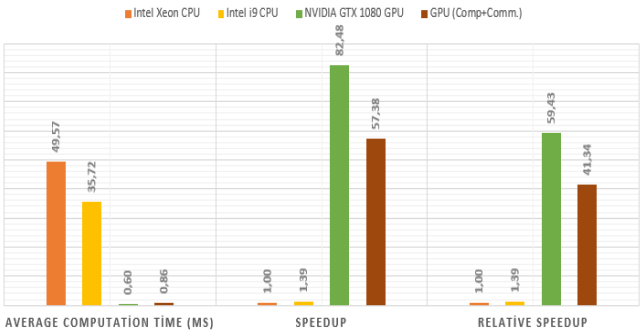


Fig. 5. Results of Time Measurements and Comparative Speedups

In view of these results, one can conclude that real-time processing of CNN-based radar detection has become possible thanks to the performance boost leveraged by graphics processors. A quarter of total GPU computation was spent on CPU-GPU data transfer. Excluding the performance loss due to data transfer, the GPU computation can achieve 60x speedup compared to the fastest CPU parallel implementation.

V. CONCLUSION

This study demonstrates parallel programming architectures for practical application of a CNN-based radar target detector with high detection performance. Thanks to the use of graphics processors and parallel programming

optimizations, real-time operation becomes possible for a sample radar system. Parallel computing with graphics processor achieves 41x times speedup compared to the highest performance CPU parallel implementation.

For future studies, the scope will be extended to cover real-time applications of new ML-based radar target detectors capable of multiple target detection.

ACKNOWLEDGMENT

This work is supported partially by the 2232 International Fellowship for Outstanding Researchers program of TÜBİTAK (Project No: 118C253). The author holds all the responsibility for the content of the paper. Financial support does not mean that the content is approved in a scientific sense by TÜBİTAK. As authors, we would like to thank Dr. Çağrı ÇETİNTEPE for his academic support.

REFERENCES

[1] O. Blumtritt, H. Petzold and W. Aspray, "Tracking the History of RADAR," *IEEE Rutgers*, New Jersey, 1994.

[2] H. M. Finn and R. S. Johnson, "Adaptive detection mode with threshold control as a function of spatially sampled clutter-level estimates," *RCA Rev.*, vol.29, 1968, pp. 141-464.

[3] Y. LeCun, Y. Bengio and G. Hinton, "Deep Learning," *Nature*, vol. 521, no. 7553, pp. 436-444.

[4] Lang, P., Fu, X., Martorella, M., Dong, J., Qin, R., Meng, X., & Xie, M., "A Comprehensive Survey of Machine Learning Applied to Radar Signal Processing", *ArXiv, abs/2009.13702*, 2020.

[5] F. Yavuz and M. Kalfa·, "Radar Target Detection via Deep Learning," *2020 28th Signal Processing and Communications Applications Conference (SIU)*, 2020, pp. 1-4, doi: 10.1109/SIU49456.2020.9302316.

[6] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, "Gradient Based Learning Applied to Document Recognition", *Proceedings of IEEE*, 86(11):2278-2324, 1998.

[7] X. Chen, Y. Qiu and H. Yi, "Implementation and performance of image filtering on GPU," *2013 4th Int. Conf. on Intelligent Control and Information Processing (ICICIP)*, Beijing, 2013, pp. 514-517.

[8] S. Chetlur, C. Woolley, P. Vandermersch, J. Cohen, J. Tran, B. Catanzaro, and E. Shelhamer, "CuDNN: Efficient primitives for deep learning," Oct. 2014, arXiv:1410.0759.

[9] F. C. Robey, D. R. Fuhrmann, E. J. Kelly and R. Nitzberg, "A CFAR adaptive matched filter detector," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 28, no. 1, pp. 208-216, Jan. 1992.

[10] A. Oppenheim and R. Schafer, *Digital Signal Processing*, pp. 585, Prentice-Hall, Upper Saddle River, N. J., 1975.

[11] P. Faraboschi, J. A. Fisher and C. Young, "Instruction scheduling for instruction level parallel processors," *Proceedings of the IEEE*, vol. 89, no. 11, pp. 1638-1659, Nov. 2001.

[12] L. Dagum and R. Menon, "OpenMP: an industry standard API for shared-memory programming," *IEEE Computational Science and Engineering*, vol. 5, no. 1, pp. 46-55, Jan.-March 1998.

[13] https://software.intel.com/content/www/us/en/develop/tools/math-kernel-library.html, Accessed on: 12.11.2021.

[14] https://en.wikipedia.org/wiki/Advanced_Vector_Extensions, Accessed on: 12.11.2021.

[15] A. Krizhevsky, I. Sutskever and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[16] https://developer.nvidia.com/cudnn, Accessed on: 12.11.2021.

[17] https://www.cpubenchmark.net/cpu.php?cpu=Intel+Core+i9-9900K+%40+3.60GHz&id=3334, Accessed on: 12.11.2021.

[18] https://www.cpubenchmark.net/cpu.php?cpu=Intel+Xe on+E5-2637+v4+%40+3.50GHz&id=2383, Accessed on: 12.11.2021.

[19] https://www.videocardbenchmark.net/gpu.php?gpu=Ge Force+GTX+1080&id=3502, Accessed on: 12.11.2021.

QuadSim: A Quadcopter Rotational Dynamics Simulation Framework For Reinforcement Learning Algorithms

Burak Han Demirbilek
Aselsan Research Center
Aselsan Inc.
Ankara, Turkey
bhdemirbilek@aselsan.com.tr

***Abstract*—This study focuses on designing and developing a mathematically based quadcopter rotational dynamics simulation framework for testing reinforcement learning (RL) algorithms in many flexible configurations. The design of the simulation framework aims to simulate both linear and nonlinear representations of a quadcopter by solving initial value problems for ordinary differential equation (ODE) systems. In addition, the simulation environment is capable of making the simulation deterministic/stochastic by adding random Gaussian noise in the forms of process and measurement noises. In order to ensure that the scope of this simulation environment is not limited only with our own RL algorithms, the simulation environment has been expanded to be compatible with the OpenAI Gym toolkit. The framework also supports multiprocessing capabilities to run simulation environments simultaneously in parallel. To test these capabilities, many state-of-the-art deep RL algorithms were trained in this simulation framework and the results were compared in detail.**

Keywords — Quadcopter Rotational Dynamics, Simulation Frameworks, Reinforcement Learning, Control Algorithms, UAVs.

I. INTRODUCTION

Deep reinforcement learning is a subfield of the domains of machine learning, optimal control and decision-making. With recent years, the theory and the applicability have been extended to use in many research areas, including intelligent control. Research and development on real-life applications needs resources such as time or money, and therefore the use of simulations is necessary to avoid such costs. In this study, the reference tracking control task of a quadcopter is investigated, and a simulation framework has been designed to allow researchers to use reinforcement learning algorithms (and as well other algorithms) in quadcopter based control research.

At the present time, there exists many quadcopter simulation frameworks such as [1], [2], [3] and they can provide many features and capabilities that are adequate to many research topics. For our research needs, we need a simple simulator where the users can start developing in minutes (aims to be plug-and-play) and every aspect of the source code can be customizable. The simulator is designed to test control algorithms for many configurations and dynamics using reinforcement learning algorithms in accordance with real-life pre-flight check procedures. The QuadSim simulation framework aims to support linear/nonlinear dynamics, deterministic/stochastic simulations, OpenAI Gym Toolkit [4] compatibility to allow many open-source algorithms to be run without any effort (such as Stable Baselines 3 [5] which has been used in the experiments section) and lastly, the single/parallel thread training sessions for high performance.

The layout of this paper is as follows: Firstly, in Section II, the problem will be defined and mathematically modeled. In Section III, the simulation framework will be explained with the design choices and experimental studies. After that the design of the performance metrics will be explained in Section IV and then the training framework, results and comparisons will be in the Section V. Lastly, the conclusion and future works will explain in the Section VI.

II. PROBLEM DEFINITION AND MODELLING

The main focus of this simulation framework is to allow users to design and develop intelligent controllers within the scope of controlling angular rates. For this reason, only the rotational dynamics of the quadcopter are taken into consideration because of the indoor applicability purposes. The real-life applicability of this simulation framework is to test quadcopters on platforms (also known as testbeds) [6] with 3 DoF (degrees of freedom) and therefore, some assumptions/limitations has been made to both simulation and real-life studies applicable and easy.

For the mathematical definition and modelling, this work is verified with the similar studies [7] in the quadcopter dynamics literature to ensure that the mathematical modelling is correct. Below, you can see the assumptions made to make sure this mathematical model is correct:

- The rotational motion of the quadcopter is independent of its translational motion.
- The center of gravity coincides with the origin of the body-fixed frame.
- The structure of the quadcopter is rigid and symmetrical, with the four arms coinciding with the body x- and y-axes.
- Drag and thrust forces are proportional to the square of the propeller's speed.
- The propellers are rigid.

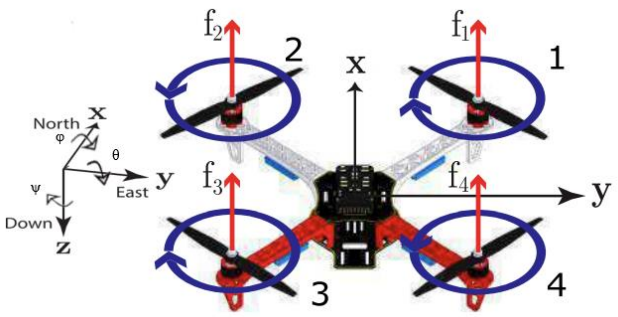


Figure 1 - Earth-fixed and body-fixed coordinate frames, motor labels and quadcopter euler angles (Adapted from [19])

The mathematical formulation starts with definition of some coordinate frames. Coordinate frames are needed to describe the motions of the quadcopter before the quadcopter is mathematically modeled.

A. Coordinate Frames

We define two coordinate frames, the earth fixed frame (notated as W) and the body fixed frame (notated as B):

The earth-fixed frame is taken as the reference frame using the NED (North East Down) convention where the x-axis of the frame is pointed to the north. The orientation of the quadcopter, known as attitude, is expressed in the body-fixed frame by Euler angles phi, theta, psi which corresponds to the roll, pitch and yaw angles. In order to relate the orientation of the quadcopter in the earth-fixed frame, a rotation matrix is defined. Below, the rotation matrix R is defined, it converts from the body-fixed frame to the earth-fixed frame (The notation of C_x *and* S_x are the abbreviated versions of the $\cos(x)$ and $\sin(x)$ operators):

$$R = \begin{bmatrix} C\psi\, C\theta & C\psi\, S\theta\, S\varphi - S\psi\, C\varphi & C\psi\, S\theta\, C\varphi + S\psi\, S\varphi \\ S\psi\, C\theta & S\psi\, S\theta\, S\varphi + C\psi\, C\varphi & S\psi\, S\theta\, C\varphi - C\psi\, S\varphi \\ -S\theta & C\theta\, S\varphi & C\theta\, C\varphi \end{bmatrix}$$

With the rotation matrix, you can calculate the earth-fixed position of any given position.

$$v' = \boldsymbol{Rv} \qquad \mathrm{v} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

After that, the body frame can be defined. A typical quadcopter is driven by four rotors on the four arms of the frame. The position of the arms according to the body frame can differ. Generally, the most used configurations are "X" and "+" which represents the shape of x and plus shape.

In this study, the "X" shape frame was used and below, you can see the numerated rotors (from 1 to 4), rotation of the numerated rotors (clockwise or counter cw) and the body frame in Figure 1. The defined roll, pitch and yaw angles (notated as ϕ, θ, ψ) are also depicted in Figure 2Figure 1 as the rotational elements on the x-y-z coordinate axes respectively.

B. Dynamics Formulation

From the rotational equations of motion, quadcopter dynamics can be formulated, for these dynamics, we firstly define the inputs as:

$$U_1 = db(w_4^2 - w_2^2)$$
$$U_2 = db\,(w_1^2 - w_3^2)$$
$$U_3 = \mathrm{k}\,(w_1^2 + w_3^2 - w_2^2 - w_4^2)$$

Where:

- b - thrust coefficient
- k - aerodrag coefficient
- d - moment arm
- w - motor speed

After that, the nonlinear system dynamics can be written from the rotational equations of motion as:

$$\ddot{\phi} = \frac{(I_{yy} - I_{zz})\theta\dot{\psi}}{I_{xx}} + \frac{U_1}{I_{xx}}$$
$$\ddot{\theta} = \frac{(I_{zz} - I_{xx})\phi\dot{\psi}}{I_{yy}} + \frac{U_2}{I_{yy}}$$
$$\ddot{\psi} = \frac{(I_{xx} - I_{yy})\dot{\phi}\dot{\theta}}{I_{zz}} + \frac{U_3}{I_{zz}}$$

For both linear and nonlinear systems, below you can find the state and input definitions:

$$x = \begin{bmatrix} \varphi \\ \dot{\varphi} \\ \theta \\ \dot{\theta} \\ \psi \\ \dot{\psi} \end{bmatrix}, \dot{x} = \begin{bmatrix} \dot{\varphi} \\ \ddot{\varphi} \\ \dot{\theta} \\ \ddot{\theta} \\ \dot{\psi} \\ \ddot{\psi} \end{bmatrix}$$

$$u = \begin{bmatrix} U1 \\ U2 \\ U3 \end{bmatrix} = \begin{bmatrix} db(w_4^2 - w_2^2) \\ db(w_1^2 - w_3^2) \\ k(w_1^2 + w_3^2 - w_3^2 - w_4^2) \end{bmatrix}$$

C. State Space(Linear) Representation Of The Quadcopter System

An LTI state space model is written below for model analysis and numerical calculations.

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du$$

The system was linearized at the hover position, and it has been proven that the system is fully observable [7].

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 & 0 & 0 \\ \frac{1}{I_{xx}} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \frac{1}{I_{yy}} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{I_{zz}} \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, D = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Where the state vector x and the input vector u is defined like this:

$$x = \begin{bmatrix} \phi \\ \dot{\phi} \\ \theta \\ \dot{\theta} \\ \psi \\ \dot{\psi} \end{bmatrix}, \dot{x} = \begin{bmatrix} \dot{\phi} \\ \ddot{\phi} \\ \dot{\theta} \\ \ddot{\theta} \\ \dot{\psi} \\ \ddot{\psi} \end{bmatrix}$$

$$u = \begin{bmatrix} U1 \\ U2 \\ U3 \end{bmatrix} = \begin{bmatrix} db(w_4^2 - w_2^2) \\ db(w_1^2 - w_3^2) \\ k(w_1^2 + w_3^2 - w_2^2 - w_4^2) \end{bmatrix}$$

D. Nonlinear Dynamics

The summarized nonlinear dynamics can be written as:

$$\dot{x} = f(x, u)$$
$$y = g(x, u)$$

$$\dot{x} = f(x, u) = \begin{bmatrix} \dot{\phi} \\ \frac{(I_{yy} - I_{zz})\theta\dot{\psi}}{I_{xx}} + \frac{U_1}{I_{xx}} \\ \dot{\theta} \\ \frac{(I_{zz} - I_{xx})\phi\dot{\psi}}{I_{yy}} + \frac{U_2}{I_{yy}} \\ \dot{\psi} \\ \frac{(I_{xx} - I_{yy})\phi\dot{\theta}}{I_{zz}} + \frac{U_3}{I_{zz}} \end{bmatrix}$$

$$y = g(x, u) = x = \begin{bmatrix} \phi \\ \dot{\phi} \\ \theta \\ \dot{\theta} \\ \psi \\ \dot{\psi} \end{bmatrix}$$

III. Simulation Framework

The framework definition is divided into 3 subsections and described in detail.

A. Simulating/Solving The ODEs

To be able to compatible with the most recent machine learning tools, python programming language was selected to implement this simulation framework. Python is a high-level general-purpose programming language which can be used in many areas, and it is designed to be readable and easy to use. Since this design is limits the code execution performance when compared it with different programming languages, many extensions of Python have been released as modules (or packages) which can be installed on the python installation.

Scipy [8] is an open-source collection of optimized mathematical algorithms for python programming language. It includes algorithms from the topics of integration, optimization, interpolation, Fourier transforms, signal processing, linear algebra, etc. In this study, the Scipy ordinary differential equations solver functions were used with the "Explicit Runge-Kutta method of order 5" solver.

Within the solver parameters, this simulation aimed to have 250Hz simulation step frequency.

B. OpenAI Gym Environment

To ease up the usage of the simulation for RL and other agents/algorithms, the simulation has been wrapped up with OpenAI Gym Toolkit [4]. Gym is a toolkit for comparing and developing reinforcement learning algorithms which is developed by OpenAI (open source). It can be used with the any numerical computation libraries such as Tensorflow [9], PyTorch [10] or Theano [11], and it contains collections of prebuilt test problems/environments which the researchers can use these environments or create their own by using the very basic class definitions. In summary, a gym environment consists of initialization, step and reset functions. Initialization function is responsible to create and initialize the environment with given parameters. A step function is the interaction function which iterates the simulation by 1 time step, returning observations/states with some other information. Reset function is to reset the simulation by setting internal variables and calling other functions, which it makes the simulation episodic.

By making simulations in accordance with these 3 basic functions, the generalized problem structure enables many algorithms to be applied into the structured simulation and with this way, the linkage between the simulation and the algorithms can be separated. For these reasons, OpenAI Gym toolkit was used in the simulation design to allow many algorithms can be run on the QuadSim environments (simulations).

1) Environment States

With the given system dynamics, the aim is to track reference angle signals(roll, pitch and yaw angles) in the bestway. Therefore, we augment the state with extra 6-dimensional information which tells what reference signals the controller must follow.

To summarize, the augmented state is defined like this:

$$x = \begin{bmatrix} \phi_{err} \\ \dot{\phi}_{err} \\ \theta_{err} \\ \dot{\theta}_{err} \\ \psi_{err} \\ \dot{\psi}_{err} \end{bmatrix}$$

Note that error variables are defined as the difference between the reference state and current state. The angle states are also mapped between [-pi, pi).

2) Environment Actions

The actions are defined in the same way as the original definition, actions are a vector with size (3,) and it tells the torque vector acting on the quadcopter in the body-fixed frame.

$$U = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix}$$

The environment will automatically calculate the propeller speeds with its motor mixing algorithm. So, the agent only needs to give the environment the torque vector.

3) Environment Limits (min and max)

After defining states and actions, the size of the state and action space needs to be defined. Since states and actions

are not limited in theory, we can limit them by specify the duration of the simulation, defining t_{start}, t_{end}.This will boundthe state and action space in terms of time duration, and limit values can be calculated when the time range is defined inthe simulation initialization.

$$Thrust = b\sum_{i=1}^{4} w_i^2 = F = m * g$$

$$w_{min} = w_i = \sqrt{\frac{m*g}{4*b}}$$

$$\mathrm{w}_{max} = \text{Can be taken from the dc motor datasheets.}$$

To convert these units of RPM (rotation per minute) to radians per second, the simple conversion equation can be used

$$rad/s = \frac{RPM * 2\pi}{60}$$

From the motor min-max speed values, we can calculate the U_{min} and U_{max} values:

$$U_{max} = \begin{bmatrix} U_1 \\ U_2 \\ U_3 \end{bmatrix} = \begin{bmatrix} db\left((\mathrm{w}_{max}^2 - \mathrm{w}_{min}^2)\right) \\ db\left((\mathrm{w}_{max}^2 - \mathrm{w}_{min}^2)\right) \\ 2k\left((\mathrm{w}_{max}^2 - \mathrm{w}_{min}^2)\right) \end{bmatrix}$$

$$U_{min} = -U_{max}$$

After that, we can calculate maximum state values when the motor speeds are constant by calculating the definite integral over this period. Please note that rotational values are already bounded by pi, so we don't need to calculate the maximum value of them.

$$\varphi_{max} = \pi\, rad$$

$$\dot{\varphi}_{max} = \frac{U_{1max}}{I_{xx}} * (t_{end} - t_{start})\, rad/s$$

$$\theta_{max} = \pi\, rad$$

$$\dot{\theta}_{max} = \frac{U_{2max}}{I_{yy}} * (t_{end} - t_{start})\, rad/s$$

$$\psi_{max} = \pi\, rad$$

$$\dot{\psi}_{max} = \frac{U_{3max}}{I_{zz}} * (t_{end} - t_{start}) rad/s$$

As these hard limits could not be reached in any real-life application, since the calculation does not include any physical limit, only the linear model is considered, additional soft limits are defined to have more accurate limitations, these limitations are used from the Pixhawk 2.1 Cube IMU Sensor(MPU-9250) Data sheet [12], which supports measurements up to 2000 degree/second (~35 radian/second)

$$\varphi_{max} = \pi\, rad$$
$$\dot{\varphi}_{max} = 35\, rad/s$$
$$\theta_{max} = \pi\, rad$$
$$\dot{\theta}_{max} = 35\, rad/s$$
$$\psi_{max} = \pi\, rad$$
$$\dot{\psi}_{max} = 35\, rad/s$$

4) Simulation Frequency

For this simulation, a simulation frequency of 250Hz and controller (control algorithm) frequency of 50Hz have been selected and used.

From the OpenAI Gym perspective, the action selected from the algorithm will be constant for every 0.02(1/50) seconds.

5) Reward/Cost Function

Since the aim of this control problem is tracking the reference signals, a reward/cost function is defined to be a quadratic cost of error, error means that the difference between the reference and current state. With such quadratic cost, the RL algorithms can be compared and criticized with the optimal control algorithms.

Below, this definition is expressed as a formal mathematical expression.

$$\overline{x} = x_{ref} - x_{current}$$

$$cost = \overline{x}^T Q \overline{x} + u^T R u$$

$$Q = identity(6,6)/\text{max state values(soft limits)}$$

$$R = identity(3,3)/\text{max action values(soft limits)}$$

$$\text{reward} = -\text{cost}$$

Where Q and R are reward/cost matrices, which are assumed to be normalized identity matrices of shape (6,6) and (3,3) respectively. The shape of the current and reference states vectors are (6,1). The term of soft limits means that soft limit values (which is explained in section 3) have been used in this normalization because of the theoretic hard limits are unrealistic and large since no physical assumptions (air friction, material properties, etc.) made about these values. In empirical experiments, it has been observed that the change in reward due to normalization is very small at low speeds, and this affects learning performance. In order to solve this problem, normalization is done by using the soft limits of the system.

6) Reset Function

In each reset, reference states of phi, theta and psi states are randomly generated in range [-pi, pi). The current state is then calculated with the difference of reference state and current dynamics state. Also, the quadcopter dynamic states are not set to reset in each call to this function. The only case for internal dynamics to be reset is when the accumulated states exceed the soft limits of the environment.

$$x_{ref} = random(6,1)$$

$$x_{current} = \text{initial states if } x_{current} \text{ exceeds soft limits}$$

$$\overline{x} = x_{ref} - x_{current}$$

Parameter	Default Value
Moment of inertia about x-axis, Ixx	0.0213 kgm^2
Moment of inertia about y-axis, Iyy	0.02217 kgm^2
Moment of inertia about z-axis, Izz	0.0282 kgm^2
Mass, m	1.587 kg
Gravity, g	9.81 N
Moment arm, d	0.243 m
Thrust coefficient, b	3.7102e-5 Ns^2
Drag coefficient, k	7.6933e-7 Nms^2
Propeller maximum angular speed, w_{max}	494.27 rad/s
Soft phidot limit, $\dot{\phi}$	35 rad/s
Soft thetadot limit, $\dot{\theta}$	35 rad/s
Soft psidot limit, $\dot{\psi}$	35 rad/s

Table 1 - Dynamic System Parameters

Parameter	Default Value
Simulation Time Per Episode	5 second
Simulation Frequency	250 Hz
Control Frequency	50 Hz
Initial State	[0, 0, 0, 0, 0, 0]
Random Seed (random seed if not set)	None
Constant Reference (random reference if not set)	None
Custom input limit (default input limits if not set)	None
Process Noise w mean and variance (only for stochastic environments)	(0, 0.01)
Measurement Noise v mean and variance (only for stochastic environments)	(0, 0.01)
Random seeds of the noises w and v (only for stochastic environment)	None

Table 2 - Environmental Parameters

7) Done Condition

Time is the only designed end condition for this simulation. This is because the simulation states and inputs are bounded for given time, so there is no need for ending the simulation earlier than planned. Parameter t_{start} and t_{end} defines how long each episode will be.

Because of the implemented systems are time invariant, this time information only defines the episode length of the simulator.

C. Configurable Parameters

The configurable parameters and their default values in the simulation framework are listed in Table 1 and Table 2:

IV. Design Of Performance Metrics

For the performance metrics, the design should cover the classical control theory performance metrics, while it also gives meaningful comparison for the learning-based methods. Below, you can see the selected metrics.

- Computation Time
- Rise Time
- Settling Time
- Overshoot Percentage
- Peak Time
- Steady State Error
- Total Cost/Reward

V. Training Framework, Results and Comparisons

To test the simulation framework, algorithm implementations from the Stable Baselines 3 [5] were used. Stable Baselines3(SB3) is a set of reliable and open-source implementations of reinforcement learning algorithms implemented by using the PyTorch [10] libraries. It's unified and tested structure makes it very easy to integrate and strong against potential bugs in the code. Below, the list of deep reinforcement learning algorithms used in testing and comparisons were shared:

- Deep Deterministic Policy Gradients (DDPG) [13]
- Proximal Policy Optimization (PPO) [14]
- Soft Actor Critic (SAC) [15]
- Twin Delayed DDPG (TD3) [16]
- Advantage Actor Critic (A2C) [17]

The learning procedure can be explained as follows. The main purpose of the learning framework is to allow agents to explore their state and action spaces effectively in order to optimize the next actions. For this aim, the learning framework should allow agents to explore many states efficiently. In Quad-Sim, the environment will randomly generate reference states in each episode and the agent will try to follow all reference signals to maximize its rewards. As a result, a generalized policy is generated to follow references for every state.

For all these algorithms, the agents were trained for 2 million time-steps to learn the reference tracking control objective. For the parameters of learning algorithms, the default parameters from the given references were used. For testing, step response control performance was measured in Figure 2. This plot was created using the QuadSim framework's default comparison function. The resulting performance metrics are also calculated and listed in Table 3.

Performance Metric / Algorithm	PID	DDPG	PPO	SAC	TD3	A2C
Computation Time (sec)	2.804	4.310	4.230	4.531	3.990	4.176
Rise Time (sec)	0.420	0.400	0.220	0.160	0.240	0.400
Settling Time (sec)	3.280	0.640	0.960	1.160	1.220	1.980
Overshoot Percentage (%)	11.97	0	0.301	61.62	6.872	18.20
Peak Time (sec)	1.300	5	0.480	0.600	0.560	0.840
Steady State Error (rad)	0.003	0.328	0.052	0.256	0.054	0.236
Total Reward (unitless)	-1.566	-14.13	-1.340	-1.922	-2.761	-3.178

Table 3 - Summary of performance metric results for trained RL algorithms. The parameters for the PID algorithm were tuned empirically and manually tuned, the performance of PID does not correspond to the optimal PID controller performance, it is used only for an example PID controller demonstration.

VI. Conclusion

In this study, a Python based quadcopter simulation framework has been designed and implemented. Then, this simulation framework has been tested with some existing state of the art deep reinforcement learning algorithms to show an example use-case. QuadSim simulation framework aims to be flexible and compatible with the existing machine learning and reinforcement learning tools. Since the main purpose of the resulting simulation framework was to study on the reinforcement learning based control algorithms, a fully customizable framework has been implemented without any code optimizations. Also, it is worth mentioning that the simulation framework code optimizations are not included in this research focus since the experiments showed that the GPU based training is more time-consuming than the simulation itself (when compared per time-step performance), therefore the simulation framework code optimizations are planned in future studies.

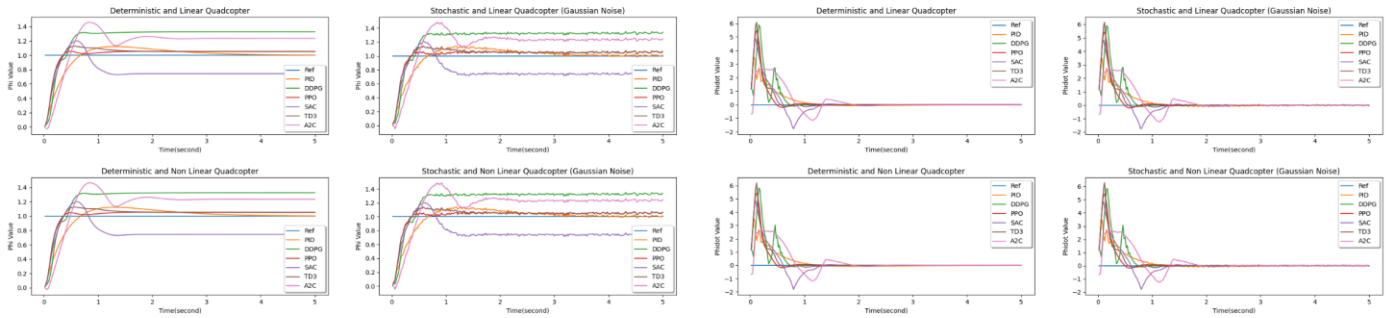


Figure 2 - The roll angle(phi) and roll angular rate (phidot) step response results of PID controller and RL algorithms DDPG, PPO, SAC, TD3, A2C. RL algorithms were trained for 2M steps with default parameters. Each plot is labeled with their environment information and their corresponding state (phi in radians and phidot in radians/sec)

As for the positive aspects of this work, after the simulation framework is downloaded, since the simulation framework is wrapped with "OpenAI Gym Environment", users will use this abstraction and they will be able to run their control algorithms without changing their own code. This will allow reaching many researchers and developers to use their own algorithms in the QuadSim simulation framework and compare their methods with others. Also, available algorithms are not limited with the reinforcement learning algorithms. Classical control and modern control theory algorithms can also be run in harmony. In addition, the simulation framework can be customized in terms of many features such as simulation dynamics, parameters, learning framework, etc.

The learning results were not examined in detail since the scope of this paper is limited with the simulation environment itself. While the resulting policy can follow the states generally, there is some steady state error in the roll state. This behavior can be fixed by increasing the Q matrix weight of the roll state while decreasing the weight of the roll angular rate state. This imperfect learning result is given by purpose to show the abilities and indicates the need for use simulation environments to improve learning performance iteratively. Further RL based controller performance and robustness will be investigated in future works.

In future studies, this framework design can be extended to have the feature of automatically executing experiments with many(like hundreds) of independent reinforcement learning training(which all of them cannot be executed in the same time because of the GPU VRAM usage of such trainings) and gathering the confidence intervals to robustly compare these algorithms with non-learning algorithms. Also, comparing the results with the most commonly used algorithms by the industry like PID, LQR, LQG, Linear MPC and Nonlinear MPC is necessary to show the performance metric results and see the real advantages/disadvantages of these RL based methods. The framework will provide such infrastructure for these studies, and it will open doors for further research.

REFERENCES

[1] S. Shah, D. Dey, C. Lovett and A. Kapoor, "AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles," in *Field and Service Robotics*, 2017.

[2] Y. Song, S. Naji, E. Kaufmann, A. Loquercio and D. Scaramuzza, "Flightmare: A Flexible Quadrotor Simulator," in *Conference on Robot Learning*, 2020.

[3] F. Furrer, M. Burri, M. Achtelik and R. Siegwart, "Robot Operating System (ROS): The Complete Reference (Volume 1)," Cham, Springer International Publishing, 2016, p. 595–625.

[4] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang and W. Zaremba, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.

[5] A. Raffin and H. et al., *Stable Baselines3*, GitHub, 2019.

[6] U. Yüzgeç, İ. Ökten, H. Üçgün, A. R. Gün, T. Türkyılmaz, M. Kesler, C. Karakuzu and U. Gökhan, "Development of the Test Platform for Rotary Wing Unmanned Air Vehicle," *Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Dergisi,* vol. 3, p. 18–24, 2016.

[7] C. A. Amadi and others, "Design and implementation of a model predictive control on a pixhawk flight controller.," 2018.

[8] P. Virtanen and G. et al., "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, p. 261–272, 2020.

[9] M. Abadi, A. Agarwal, P. Barham and E. B. et al., *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*, 2015.

[10] A. Paszke and e. a. Gross, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Advances in Neural Information Processing Systems 32*, 2019, p. 8024–8035.

[11] Theano Development Team, "Theano: A Python framework for fast computation of mathematical expressions," *arXiv e-prints*, vol. abs/1605.02688, May 2016.

[12] TDK-InvenSense, *MPU-9250 Nine-Axis (Gyro + Accelerometer + Compass) MEMS MotionTracking™ Device,* 2021.

[13] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[14] J. Schulman, F. Wolski, P. Dhariwal, A. Radford and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[15] T. Haarnoja, A. Zhou, P. Abbeel and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*, 2018.

[16] S. Fujimoto, H. Hoof and D. Meger, "Addressing function approximation error in actor-critic methods," in *International Conference on Machine Learning*, 2018.

[17] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International conference on machine learning*, 2016.

[18] G. Van Rossum and F. L. Drake, Python 3 Reference Manual, Scotts, Valley: CreateSpace, 2009.

[19] I. Grujic and R. Nilsson, "Model-based development and evaluation of control for complex multi-domain systems: Attitude control for a quadrotor uav," *Technical Report Electronics and Computer Engineering*, vol. 4, 2016.

Comparison of Activation Functions and Network Types on the Performance of Compressed Domain Action Recognition Framework

Hüseyin Onur Yağar
Informatics Institute, Ayazaga Campus
Istanbul Technical University
Istanbul, Turkey
yagar19@itu.edu.tr

Behçet Uğur Töreyin
Informatics Institute, Ayazaga Campus
Istanbul Technical University
Istanbul, Turkey
toreyin@itu.edu.tr

***Abstract*—The versatility and success of compression algorithms for videos are rapidly increasing. Compressed videos occupy less space to store more data. Hence, compression algorithms are applied almost every videos. Information extraction and interpretation of compressed domain components are more crucial than ever. In this paper, we addressed the issue of learning compressed video components. We presented an application of fire detection in compressed domain. We also analyzed different activation functions which would be better for compressed domain. Experimental results showed algorithm can successfully detected and classified fire even though there are small data.**

Keywords—Deep learning, compressed domain, fire detection, activity classification

I. INTRODUCTION

Today, there is a rapid increase in data. Since the early millennium, a hard-to-deal mass of knowledge has been growing. While some of these data are verbal and auditory, some are in the form of images. The rapid acquisition of data has led to the development of fast and automated systems for data processing. Some of the most successful examples of these systems are seen in the field of image processing. Image processing can be done over a single image frame as well as over sequential multiple images. All data that can be used in processing an image is restricted to that image itself. In videos, there are more information than one still image due to multiple images. Also, an additional information arises from the relative situation of images. Activity recognition and classification comes out exactly at this point. Understanding the transformation of an activity over time is the main problem. The offered solutions for this problem can be grouped under different areas. One of these areas uses hand-crafted features to extract necessary information to recognition. The other one uses deep learning methods to extract features automatically. We used deep learning methods to extract features needed to classify videos.

Wu et. al proposed an architecture which uses three different compressed video components to train three different neural networks for action recognition [1]. Another research transforms time domain to frequency domain. ResNet50 is modified for working in frequency domain and this network is trained with DCT coefficients. Motion vectors are trained with ResNet18 and these two networks' outputs are lately fused [2]. Multiteacher knowledge distillation is a recent technique in Deep learning area. M. Wu and C. Chiu proposed a framework for action recognition. The model is compressed by transferring knowledge from multiple teachers to a single small student model. Their experiments show that they reached a 2.4×compression rate in number of parameters and 1.2×computation reduction with 1.79% loss of accuracy on the UCF-101 dataset [3]. There is also another teacher-student method research which mimics the raw domain. Their hypothesis is that the residual component in the P-frames can fill the role of the missing RGB frames. Hence instead of using N RGB frames they used only one I-frame and N−1 P-frames. Their research accelerated action recognition in compressed domain and obtained %93.2 accuracy rate [4]. Efficiency in computation is another important topic for deep learning research. Huo et al. proposed a model that uses a novel trilinear pooling module for different type of frames. Also, their method includes training of B-frames [5]. Li et al. proposed a novel Slow-I-Fast-P (SIFP) neural network model for compressed video action recognition. It consists of the slow I pathway receiving a sparse sampling I-frame clip and the fast P pathway receiving a dense sampling pseudo-optical flow clip. An unsupervised estimation method and a new loss function are designed to generate pseudo-optical flows in compressed videos. Their model eliminates the dependence on the traditional optical flows calculated from raw videos [6]. Their team reached an accuracy rate of %94. Toreyin et al. proposed an algorithm to detect moving object and region in videos that is compressed using wavelet transform. The model predicts transform of wavelet of the background scene from the videos' past image frames' wavelet transform [7]. Savcı et al. proposed an algorithm that detects fire using macroblock types and Markov Model in H264 video. Their proposed model detects especially fire flicker process [8].

Our contribution has three folds. We used same architecture presented in [1]. Firstly, we proposed that their backbone can be replaced with computation-efficient ResNeXt50. Secondly, we analyzed different activation functions to fit better on compressed domain components. Finally, we applied our knowledge of activity classification to fire detection problem in compressed domain. We presented that algorithm successfully classified videos from each other which contains fire and smoke.

The rest of paper organized as follows. In section II, we present data types in compressed videos and datasets that we used for experiments. In section III, we show our methods. We compared neural networks to find optimum model. Then we present activation functions that used in analyzing phase.

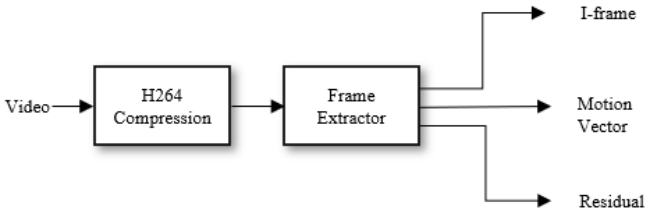


Fig. 1. Schematic structure of extraction process of compressed video components.

process. After that we show our application of fire detection model in compressed domain. In section IV, we present the experimental results. Finally, in section V we draw our conclusion.

II. DATA

In this section we present all data that we use in our studies. First, we show frame types of compressed videos in detail. Second, dataset using for action recognition is presented. Then dataset that used for fire detection in compressed domain is shown.

A. Frame Types in Compressed Videos

There are three frame types in compressed videos: I-frame, P-frame and B-frame. Fig. 1 presents schematic structure of extraction process of these frames. I-frame or intra coded frame is a complete image like JPG or BMP file. I-frame stores all pixels and information of still image. P-frame contains differences between instant frame and preceding frame. If a person is walking in front of a constant background, only motion information of person is needed to be coded. Because background information is already stored in I-frame. If we divide the P frames into two main components, the motion vectors represent blocks of pixels that change from the source frame to the target frame. Although block movements are kept in the motion vector, there may be differences between estimated frame and the original raw frame as a result of compression. These differences are represented by residuals. Finally, B-frame stores the difference between the instant frame and the previous and next frame. Thus, it takes up less space.

B. Datasets

UCF-101 was chosen as the dataset for training and testing the models during study. The UCF-101 dataset was collected from the YouTube site of realistic videos. It is an action recognition dataset containing 101 action categories. This dataset is an extended version of UCF-50 dataset. UCF-101 is a diverse dataset with differences such as appearance, pose, object scale, background, lightning conditions. Unlike other datasets which are artificial and acted by actors, UCF-101 is realistic and more challenging. The dataset consisting of 101 categories can also be divided into 25 more groups. And each group contains 4-7 motion videos. Videos that belonging to the same group may have common features such as background and angle of vision [9].

Dataset for fire detection consists of videos that made by Çetin et al. after their studies on fire-smoke detection. This dataset contains 28 different videos. The half of the videos comprise fire elements, the other half does not contain any

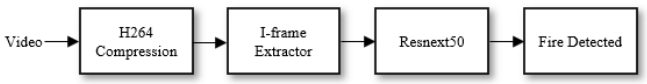


Fig. 2. Schematic structure of our fire detection application from video to detection result.

fire but smoke. Videos have diversity in terms of fire. Either huge fire spreading on forests or little fire that set in control contained by videos [10].

III. METHOD

In this section all methods are shown in scope of our research. Firstly, we compared different neural networks for computation-efficient model which can replace backbone of [1]. Secondly, we analyzed activation functions to find which one is fitting better to compressed video components. Finally, we presented our application of fire detection for compressed videos.

A. Comparison of Neural Networks

In the first part, experiments were carried out by modifying the data used in [1]. The original architecture used three different frame types and three neural networks. We concatenated three frame types to make one solid element. Unlike the original architecture we only used this element to reach more efficient model. For this reason, the data loader section has been modified to take three different frame types and convert them into a single element.

In another study we carried out to reduce the computational cost is changing backbone model with computation-efficient model. The backbone of [1] is ResNet152 which has got 58 million parameters. To replace this model, low-consumption models are trained such as ResNeXt50, MobileNetV2 and MNASNet1.0. These models contain 23 million, 2.3 million and 6.2 million parameters respectively. When searching for an optimum model, trainings were carried out on I-frames.

B. Activation Functions

In this part of the research, a comparison-analysis study was carried out over the activation functions. Activation functions are one of the important components of deep learning models. During forward propagation, the activation function makes the node non-linear and decides whether the neuron should pass information. The output of the neuron changes according to the size of the signal allowed by the activation function. In addition, the output of the neuron is normalized between 0 and 1 or -1 and 1, depending on the type of activation function used. The activation function directly affects the performance of the model and the computational costs. Because the activation functions are related to the convergence speed of the model. Also, activation function has an important place in terms of processing and transmitting information between deep learning layers and this affects model performance, too. Activation functions should be differentiable. Thus, the weights of neurons in the model can be updated during backpropagation to reduce the error rate.

These information about activation functions show us that one can use different activation functions for different tasks. The original architecture of ResNeXt uses rectified linear unit

TABLE I. COMPARISON OF NEURAL NETWORKS TO FIND AN OPTIMUM MODEL REPLACING BACKBONE RESNET152

Model Name	Parameter Number	Accuracy Rate
ResNet152	58.350.757	%88.40
ResNeXt50	23.186.253	%84.96
MobileNetV2	2.353.2536	%78.08
MNASNet1.0	6.231.693	%74.39

(ReLU) as activation function. Scope of our research we carried out different trainings with other activation functions to find a better activation function for compressed video components. We performed 6 trainings with 5 different activation functions. The functions ReLU, leaky rectified linear unit (LeakyReLU), rectified linear unit 6 (ReLU6), Mish, scaled exponential linear unit (SELU) are used for training of ResNeXt50 on I-frame.

C. Fire Detection in Compressed Videos

In the last part of our research, we presented a solution to fire detection in compressed videos and schematic structure of our application can be seen in Fig. 2. A large number of fires occur in the world every year. Fires, both inside and outside of buildings, are extremely dangerous for life. Early detection of fire is key importance to prevent fire. There are many different fire detection methods available. Fire, smoke detectors can be thought as a first precaution to prevent fire. However, these detectors cannot promise an effective operation when it comes to open area. Instead, computer vision systems can be used in open areas, forest, etc. for fire detection. We propose that compressed video components namely I-frame can be used for fire detection in videos. We used deep learning method to extract features from I-frames and ResNeXt50 was our model. In our algorithm I-frames are directly given to neural network for training. Our model output shows classification of videos to either fire or nonfire. We firstly divided videos two equal separate categories fire and nonfire. Then 20 videos were prepared for training and 8 videos for test. Videos were compressed with H264 format. Then I-frame components were extracted compressed videos and trainings were carried out.

IV. EXPERIMENTAL RESULTS

In the beginning of our experiments, we concatenate three different compressed frame types namely I-frame, motion vector and residual to one solid element. We performed a training with ResNet152 on this element. After training %78 accuracy was reached. But for classification task, this result is not promising. While original architecture reaches %90 accuracy level.

When we used other models to replace backbone ResNet152 as our next experiment, we trained models on I-frames. We used UCF-101 action video dataset and compressed these videos with H264 format. Then we extracted I-frame of these compressed videos. Three models were used for searching of an optimal model in terms of computation and success. Result of this experiment can be seen in Table I. ResNet152 presented %88.94 accuracy rate. When training was carried out with ResNeXt50 which has got almost half number of parameters than ResNet152, %84.96 accuracy rate was achieved. ResNet152 required 3 days of training phase while ResNeXt50 reached end of training after one and a half day. Other models MobileNetV2 and MNASNet1.0 reached accuracy rates of %78.08 and %74.39 respectively. Due to very low-number of parameters these models have, obtained results were looked rational. However, assumption of more parameters are prone to more accuracy rate is not true.

TABLE II. COMPARISON OF ACTIVATION FUNCTIONS IN SEARCH OF BEST FITTING FUNCTION TO COMPRESSED DOMAIN VIDEO

Activation Function	Negative Slope	Accuracy Rate
ReLU	-	%85.88
LeakyReLU	0.01	%85.2
LeakyReLU	0.1	%83.9
ReLU6	-	%85.09
Mish	-	%71.69
SELU	-	%55.91

Some activation functions can be prone to solve specific problems. We carried out 6 neural network trainings for 5 different activation functions. As a result of these trainings, it has been observed that the activation functions related to ReLU family are more effective. The ReLU activation function had the highest accuracy rate as all results are showed in Table II. LeakyReLU function has a parameter of negative slope which determines inclination of function on the negative side of origin. This slope parameter value was defined as 0.1 and 0.01. After these operations two trainings were performed and obtained results were similar to ReLU function. Also, after training which used ReLU6 activation function, similar results were obtained again with the training which ReLU was used in. The Mish function which has been frequently mentioned on research, was insufficient for the given task. It would be better not to use the Mish function for this task, which achieves better results than ReLU in many different tasks. SELU, the other activation function, showed a poor performance for the defined classification process. All this trainings and experiments showed us that ReLU activation functions is a good choice for learning features of compressed video components.

For our last experiment, videos in fire detection dataset were divided to equal groups. Then all videos were compressed with H264 format. After extraction of I-frame from these compressed videos, training was performed. ResNeXt50 and ReLU were used in training as the neural network and activation function. Input of ResNeXt50 of our model was 224x224. We fed our model with I-frames. Model outputs probability distribution of fire and nonfire categories for every I-frame. For a single video, we took mean of these probabilities to assign video to one class. We reached best accuracy rate on our 151 epochs of training. Our algorithm successfully classified videos with %87.5 accuracy rate.

V. CONCLUSION AND FUTURE WORK

In this paper we presented our analysis on neural networks and application of an algorithm on fire detection. We showed that heavy neural network of ResNet152 can be replaced with rather lightweight counterpart ResNeXt50 while little dropping on accuracy rate. This replacement presented important efficiency in terms of computation. Then we made a comparison-analysis on activation functions to search of a well-suited function for compressed video components. In our experiments, we used some of the most preferred activation functions in computer vision field. Our studies showed that ReLU activation function was superior to

other activation functions. Even though last research showed Mish activation function as a well replacement ReLU, our studies presented that ReLU was a good choice of activation function for compressed video components.

In our last experiments we made an application of fire detection system. Number of compressed videos is increasing with the success of compression algorithm on videos. So, we presented that compressed videos can be used for fire detection. After extraction of I-frame components of compressed video, we used deep learning method to detection of fire in videos with I-frame. For the future work we plan to extend our research on using discrete cosine transform (DCT) coefficients of compressed videos to detect fire.

ACKNOWLEDGEMENT

This work is supported in part by TÜBİTAK-ARDEB 1001 Program with grant number 121E378.

REFERENCES

[1] Wu, C., Zaheer, M., Hu, H., Manmatha, R., Smola, A., Krahenbühl, P. "Compressed Video Action Recognition". IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2018.

[2] Santos S. F. d. & Almeida J. "Faster and Accurate Compressed Video Action Recognition Straight from the Frequency Domain". 33rd SIBGRAPI Conference on Graphics, Patterns and Images. 2020.

[3] Wu M., Chiu C. & K. Wu. "Multi-teacher Knowledge Distillation for Compressed Video Action Recognition based on Deep Learning". IEEE International Conference on Acoustics, Speech and Signal Processing. 2019.

[4] Battash B., Barad H., Tang H. & Bleiweiss A. "Mimic The Raw Domain: Accelerating Action Recognition in the Compressed Domain". IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. 2020

[5] Huo, Y., Xu, X., Lu, Y., Niu, Y., Ding, M., Lu, Z., Xiang, T., & Wen, J." Lightweight Action Recognition in Compressed Videos". ECCV Workshops. 2020.

[6] Li J., Wei P., Zhang Y. & Zheng N. "A Slow-I-Fast-P Architecture for CompressedVideo Action Recognition". Proceedings of the 28th ACM International Conference on Multimedia. 2020.

[7] B. U. Toreyin, A. E. Cetin, A. Aksay and M. B. Akhan, "Moving region detection in wavelet compressed video". Proceedings of the IEEE 12th Signal Processing and Communications Applications Conference. 2004

[8] M. M. Savcı, Y. Yıldırım, G. Saygılı and B. U. Töreyin, "Fire Detection in H.264 Compressed Video". .ICASSP - IEEE International Conference on Acoustics, Speech and Signal Processing. 2019.

[9] K. Soomro, A. R. Zamir, and M. Shah. "UCF101: A dataset of 101 human actions classes from videos in the wild". CRCV-TR-12-01. 2012.

[10] signal.ee.bilkent.edu.tr/VisiFire/

Development of a Digital Twin Model for a Stepper Motor under Mechanical Strain

Elif Cesur
Industrial Engineering
Istanbul Medeniyet University
Istanbul, Turkey
elif.karakaya@medeniyet.edu.tr

Muhammet Raşit Cesur
Industrial Engineering
Istanbul Medeniyet University
Istanbul, Turkey
rasit.cesur@medeniyet.edu.tr

***Abstract*— Recently, production systems have become more dynamic and intelligent because the reality of mass customization emerged as a result of industrial transformation. In addition to changeability in customer requirements, communication technologies that make a digital copy of a real system possible have improved. At this point, a new modeling approach came to exits called Digital Twin (DT) to evaluate a system result before it happens. Mass customization, which is one of the results of industrial transformation, has forced the implementation of more dynamic and intelligent production systems. On the other hand, with the advances in communication technologies, it was aimed to make a digital copy of real production systems and it was shown that the expected results could be calculated with the help of Digital Twin DT). It is aimed to create a digital twin of a stepper motor under mechanical strain. During the generating DT phase, data were collected by measuring the execution time of the CNC workbench's motion commands. In this study, different states of current levels are categorized to detect whether unexpected situations like mechanical strain occurs or not with the help of a binary classification method using an Artificial Neural Network.**

Keywords*— *Digital Twin, Smart Manufacturing, Artificial Intelligence, Mechanical Strain

I. INTRODUCTION

The development of sensor and information and communication technology (ICT) leads to a digital revolution in the manufacturing environment. The digital transformation enables widespread customization, resulting in more dynamic manufacturing processes. New generation production systems need more flexible planning and control algorithms due to their dynamism. As a result, all possible states in the production environment must be produced electronically by the planning and control system, which acts as a digital replica of the real production system. The Digital Twin (DT) is a digital replica of a real-world production system that can simulate all states in a real-world production environment and anticipate the outcomes of all transactions. With a digital twin technology developed in this study, stepper motors will be able to detect and analyze this situation in the face of an unexpected force. By detecting this situation, more efficient and effective operation of the engine will be ensured.

II. LITERATURE REVIEW

Many digital twin studies have been done, recently. The focus of the studies is to solve the operating mechanisms of the systems. Hybrid methods, statistical methods, conventional artificial intelligence, deep learning, and dynamic data modeling techniques are used to build digital twin models [1]. However, similar problems can be solved without using the digital twin. The contribution of the digital twin is to provide data for a wide range of analyses [2] tried to imitate the dynamics of the DC engine with Kalman filters and also [11] used digital twin technology for predictive maintenance. [12], [13] and [14] use digital twin technology for production planning. [15] and [16] use digital twin technology to analyze and propose a smart production system.

Because digital twin technology focused on modeling the dynamics of systems components of manufacturing should be aimed to be modeled with digital twin to achieve building a complete and complex digital twin model. One component of manufacturing systems is cartesian axes CNC workbenches, and one main part of the workbench is the stepper motor. [4] measured not only mechanical failures with 83.45% (bearing failures 80.85%, others 2.60%) but also electrical failures 16.55% (winding) by using stepper motor conveyor system. [5] figured out a rotating speed variation model with Kalman filters. Various sensors are utilized to build mentioned the digital twin models Such as [6] used load cell, temperature sensor, and the current sensor for dc driver. [7] the measured magnetic field of punch, [8] investigated vibration at gearbox of dc engine. [9] is developed a thermal model of a motor to detect faulty conditions. [10] used gyroscope and an acceleration sensor for predictive maintenance.[11] noticed the importance of vibration frequency and temperature for classifying states of the engine.

III. THEORICAL BACKGROUND

DT models of CNC machine sub-systems have been presented in several publications in the scientific literature for forecasting task outcomes and performance metrics before they begin processing. [6] [10] created DT models using artificial intelligence (AI) or machine learning methods. [6] employed Gray Relations (GR) to assess the likelihood of a malfunction on the workbench, emphasizing that Deep Learning (DL) would be more useful in the suggested model. [10] classified machine equipment as a level of availability using the Support Vector Machine (SVM) method. [11] investigated quality control using DT by attaching an accelerometer and gyroscope to the CNC tool-head. The sliding window approach is used to match sensor input with solid model data. AI is used to examine matched data. The role of AI in the aforementioned investigations, which are attempting to develop a DT model of manufacturing equipment, is to infer the equipment's functioning mechanism and provide the desired results. In this paper, we create a DT model that predicts the behavior of current drawn by a stepper motor with and without mechanical strain. Data collection is the initial step, followed by data pre-processing to train an AI model. The AI training is the third procedure. The AI model

is validated as the last step, as seen in Figure 1. Many DT research in the scientific literature uses the described methodologies for DT model creation as a framework.

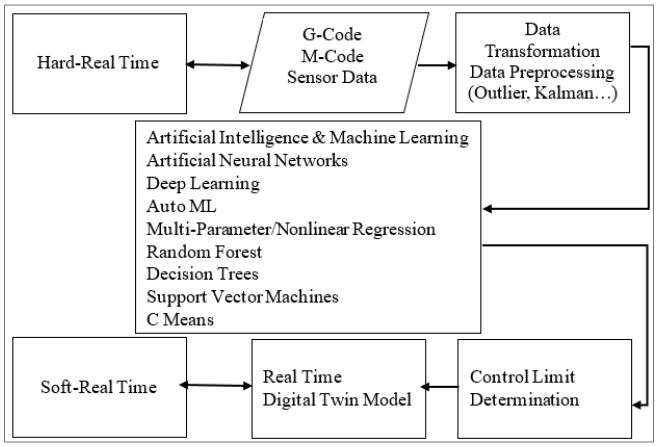


Fig. 1. DT Framework

IV. Case Study

The malfunctions of stepper motors, which are the most common motor type used in Cartesian axis machines, can be listed as driver errors, by-pass, electromagnetism problems, and problems due to mechanical stress. Failures in motors mostly affect the amount of current drawn by the motor. However, it may not always be possible to detect the problem directly with current measurement, because the amount of drawn current varies as seen in Figure 1. The first and the second peaks are plots of the current under half speed. The fourth, sixth, and eighth peaks are plots of the current under full speed. The third, the fifth, and the seventh peaks are the plot of the current under mechanical stress. As it is seen in Figure 2, plots of peak points, whose data is collected under the same conditions, are different visually.

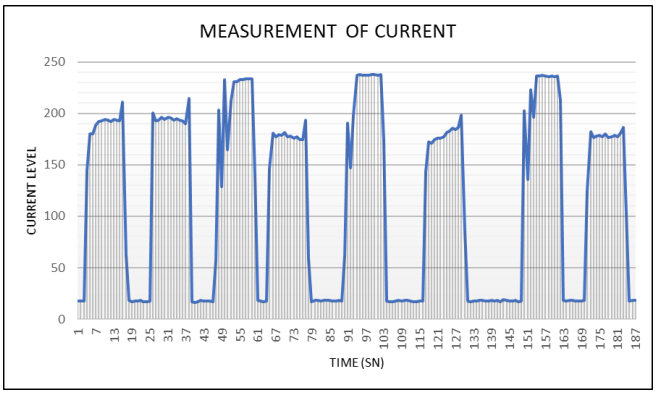


Fig. 2. Observed current data from the current sensor

There is a need for a more deterministic parameter to classify the state of the motor, which is more relative than the direct measurement of the current value with the change of state. The amount of change in current values, seen in Figure 3, is investigated whether having a strong correlation with the state of the stepper motor, or not. There is an obvious multiple traverse shape on the current plot when mechanical stress is applied to the stepper motor. When all traverses are investigated, eight states are determined related to motor characteristics. Within the scope of this study, 8 states were defined. These are;

1) Idle state of stepper motor
2) Beginning of motion
3) Being in motion
4) Finalize in motion
5) Beginning to move under mechanical strain
6) Beginning of mechanical strain
7) Being of motion under mechanical strain
8) Finalize in motion under mechanical strain

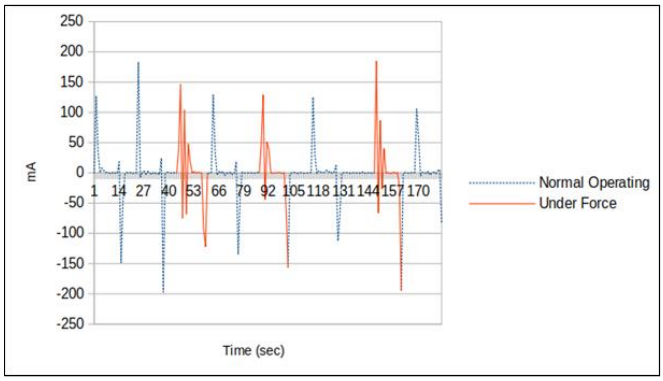


Fig. 3. Change of motor current value

A deep neural network with four hidden layers is constructed to perform the binary classification task. An average of four sequential current measurements and three sequential change amounts of the four current measurements are used as inputs to classify traverses. The hidden layer consists of four sub-layers, which are 300 nodes dense layer, 400 nodes dense layer, 200 nodes dense layer, and 50 nodes dense layer. Finally, the output layer contains 8 nodes for binary classification. Every node represents one state of the stepper motor. Softplus is used as an activation function at every layer. 20% of data is used for validation. Mean squared error(MSE) is used as a loss function, and Adam Optimizer is used for training. 7500 epocs are found as the optimal number of training iterations as seen in Figure 4.

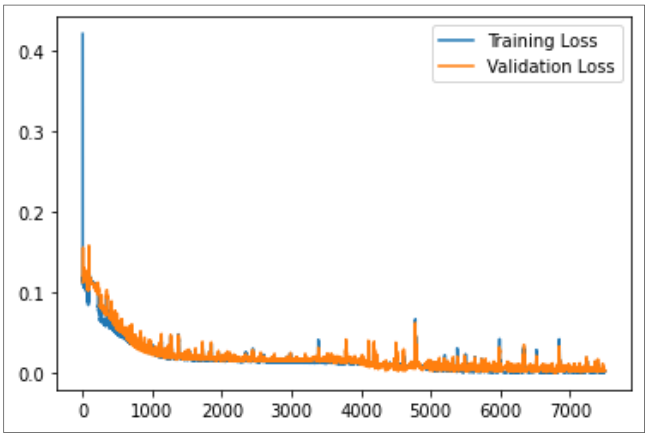


Fig. 4. Change of loss function during training

V. Conclusion

During the industrial revolution, the DT technique was first used to build a dynamic and adaptable production system. However, both the literary and the industrial sectors lack sufficient theoretical and practical applications. At this point, this study aims to develop a digital twin model, which will enable predictive maintenance of stepper motors, which are frequently used in Cartesian machines to understand their operating states and to report abnormal situations. To achieve this, deep learning was used and the amount of change in the current values of the motor was evaluated and the classification of the state of the motor was provided. In this way, it has become possible to analyze energy consumption, efficiency, and reliability depending on the operating conditions of the engine. The contribution of digital twin technology is to understand whether the engine is operating normally depending on the starting speed. In the next study, it is aimed that the developed model will be associated with the commands sent to the machine, and it will be able to predict and classify the reactions of the engine depending on the production parameters during the production of a part that has not yet been produced.

References

[1] Y. Wen, M. Fashiar Rahman, H. Xu, and T.-L. B. Tseng, "Recent advances and trends of predictive maintenance from data-driven machine prognostics perspective," *Measurement*, vol. 187, no. September 2021, p. 110276, 2022.

[2] S. K. Yang and T. S. Liu, "State estimation for predictive maintenance using Kalman filter," *Reliab. Eng. Syst. Saf.*, vol. 66, no. 1, pp. 29–39, 1999.

[3] Y. GÜLTEPE, "Makine Öğrenmesi Algoritmaları ile Hava Kirliliği Tahmini Üzerine Karşılaştırmalı Bir Değerlendirme," *Eur. J. Sci. Technol.*, pp. 8–15, Aug. 2019.

[4] A. Carignano, A. Di Nisio, A. M. L. Lanzolla, M. Savino, V. L. Scarano, and M. Spadavecchia, "Methodological proposal for Accelerated Screening Test design on stepper motors," *Meas. J. Int. Meas. Confed.*, vol. 54, pp. 241–248, 2014.

[5] S. K. Yang, "An experiment of state estimation for predictive maintenance using Kalman filter on a DC motor," *Reliab. Eng. Syst. Saf.*, vol. 75, no. 1, pp. 103–111, 2002.

[6] M. Mazzoleni, M. Scandella, F. Previdi, and G. Pispola, "Data on the first endurance activity of a Brushless DC motor for aerospace applications," *Data Br.*, vol. 29, p. 105153, 2020.

[7] H. A. Weiss, N. Leuning, K. Hameyer, H. Hoffmann, and W. Volk, "Manufacturing efficient electrical motors with a predictive maintenance approach," *CIRP Ann.*, vol. 68, no. 1, pp. 253–256, 2019.

[8] R. Zhang, F. Gu, H. Mansaf, T. Wang, and A. D. Ball, "Gear wear monitoring by modulation signal bispectrum based on motor current signal analysis," *Mech. Syst. Signal Process.*, vol. 94, pp. 202–213, 2017.

[9] M. J. Picazo-Ródenas, R. Royo, J. Antonino-Daviu, and J. Roger-Folch, "Use of the infrared data for heating curve computation in induction motors: Application to fault diagnosis," *Eng. Fail. Anal.*, vol. 35, pp. 178–192, 2013.

[10] C. Kammerer, M. Gaust, M. Küstner, P. Starke, R. Radtke, and A. Jesser, "Motor Classification with Machine Learning Methods for Predictive Maintenance," *IFAC-PapersOnLine*, vol. 54, no. 1, pp. 1059–1064, 2021.

[11] K. Mykoniatis, "A real-time condition monitoring and maintenance management system for low voltage industrial motors using internet-of-things," *Procedia Manuf.*, vol. 42, no. 2019, pp. 450–456, 2020.

[11] H. Zhang, Q. Yan, and Z. Wen, "Information modeling for cyber-physical production system based on digital twin and AutomationML," Int. J. Adv. Manuf. Technol., vol. 107, no. 3–4, pp. 1927–1945, 2020.

[12] C. Zhuang, J. Liu, and H. Xiong, "Digital twin-based smart production management and control framework for the complex product assembly shop-floor," Int. J. Adv. Manuf. Technol., vol. 96, no. 1–4, pp. 1149–1163, 2018.

[13] K. T. Park, D. Lee, and S. Do Noh, "Operation Procedures of a Work-Center-Level Digital Twin for Sustainable and Smart Manufacturing," Int. J. Precis. Eng. Manuf. - Green Technol., vol. 7, no. 3, pp. 791–814, 2020.

[14] Y. Wang, W. Ren, Y. Li, and C. Zhang, "Complex product manufacturing and operation and maintenance integration based on digital twin," Int. J. Adv. Manuf. Technol., 2021.

[15] K. T. Park, J. Lee, H. J. Kim, and S. Do Noh, "Digital twin-based cyber physical production system architectural framework for personalized production," Int. J. Adv. Manuf. Technol., vol. 106, no. 5–6, pp. 1787–1810, 2020.

[16] H. Zhang, G. Zhang, and Q. Yan, "Digital twin-driven cyber-physical production system towards smart shop-floor," J. Ambient Intell. Humaniz. Comput., vol. 10, no. 11, pp. 4439–4453, 2019.

A Survey on Turkish Named Entity Recognition

Oğuzhan Özçelik
Aselsan Research Center
ASELSAN
Ankara, Türkiye
ogozcelik@aselsan.com.tr

Çağrı Toraman
Aselsan Research Center
ASELSAN
Ankara, Türkiye
ctoraman @aselsan.com.tr

***Abstract*— Named entity recognition is a challenging task that detects important entities in text; such as person, location, and organization. There are many efforts to solve this task, specifically in English text. However, the results of named entity recognition methods could be different for Turkish text due to its agglutinative structure. In this study, we provide a detailed survey on the studies that detect named entities in Turkish text. We divide the current studies into rule, statistical, neural network, and Transformer-based solutions. We then provide up-to-date state-of-the-art scores for Turkish named entity recognition, along with a brief discussion on the main challenges in Turkish named entity recognition.**

Keywords— named entity recognition, survey, Turkish text

I. INTRODUCTION

Named Entity Recognition (NER) is an essential subtask of information extraction, which classifies and tags the predetermined named entities in a text. These named entities can differ according to domains. Although named entities such as Person, Location and Organization (PLO) are the most used ones in general domain, they can be diversified into fine-grained entities, such as Genes and Proteins for specific domains of Biomedicine.

Since NER is a well-studied topic for English, the reported classification accuracies are very high (>90%). However, these results are usually obtained by testing the models on formal datasets such as news articles and books, where less lexical and syntactical errors are observed. On the other hand, today, excessive use of social media creates demand on research for noisy informal data collected from social media platforms, such as Twitter, where it includes lots of spelling errors, abbreviations, semantic ambiguity, user generated words and so on. This makes informal texts difficult to analyze, which results in lower scores on such noisy social media datasets. Specific methods like a text normalization or social media domain-specific features may be needed for these tasks. From these perspectives, NER is still an active research topic that needs to be considered and investigated.

In this paper, we examine NER for Turkish language. Turkish is a morphological rich language, i.e. thousands of words can be generated from a single word by adding new suffixes and that word contains lots of information. For instance, “evdeyseniz” (translated into: “If you are at home”) is a single word includes locative (-de), conditional (-yse), plural (-niz) information. Moreover, Turkish is a free order language, i.e. words’ positions can be changed freely without losing semantic meaning of the sentence. For example, “Mustafa Kemal Atatürk 1881’de doğdu.” and “1881’de, Mustafa Kemal Atatürk doğdu.” are the same in semantics, translated into “Mustafa Kemal Atatürk was born in 1881.”. Therefore, such grammar structures make NER a challenging task in Turkish.

This is a brief survey for Turkish NER, where methods and challenges are investigated for Turkish NER. Our contribution is to provide an up-to-date brief survey for Turkish NER. In Turkish NER studies, BERTurk-CRF architecture achieves state-of-the-art result in [2], on a formal dataset News Article [1]. For informal, Twitter Dataset [3], BiLSTM-CRF architecture employing transfer learning achieved state-of-the-art result in [4], where source is trained on News Articles, and the target is Twitter Dataset.

The rest of the paper is organized as follows. Section II gives background information on tagging schemes and evaluation metrics. Section III describes methods and studies in Turkish NER literature. Section IV gives brief discussion on methods and challenges. Section V summarizes this survey and provides future works.

II. BACKGROUND

There are different tagging schemes and evaluation metrics in NER. These are important while preparing the dataset and evaluating the results of the models and comparing their result with other studies. Although these tagging schemes and metrics have relatively small differences between them, it is important to compare results under the same metrics to have a meaningful outcome.

A. Tagging Schemes

There are three common tagging schemes: IOB1, IOB2, IOBES; where B, I, O, S, and E simply refers to beginning, inside, outside, single and end, respectively. The example outputs of a given sentence are given in Table I.

1) IOB2

IOB2 is the most common used tagging scheme. In the IOB2 scheme tagging, the first token of the named entity is tagged as (B-type), where “type” refers to the corresponding named entity, such as ‘PERSON’, ‘LOCATION’, etc. Remaining tokens of the named entity are tagged with (I-type), if there exist any.

TABLE I. TAGGING SCHEMES

Words	**Tagging Schemes**		
	IOB2	***IOB1***	***IOBES***
Orhan	B-PER	I-PER	B-PER
Veli	I-PER	I-PER	I-PER
Kanık	I-PER	I-PER	E-PER
Oktay	B-PER	B-PER	B-PER
Rıfat	I-PER	I-PER	E-PER
Garip	B-ORG	I-ORG	S-ORG
Akımı	O	O	O
şairleridir	O	O	O

All other tokens, that do not belong any kind of named entity, are labeled with 'O'. In sentences, there might be many named entities belonging to the same type. Therefore, this scheme separates them by labeling the beginning word of each entity type with (B-type).

2) IOB1

The only difference of IOB1 from IOB2 is about the use of (B-type) tagging. In IOB1, (B-type) is used only for separating the same class of entity chunks in series. The result of the same example sentence with IOB1 in comparison with IOB2 can also be observed on the words "Orhan" and "Garip" in Table I.

3) IOBES

IOBES is an extension of IOB2. The use of I, O and B are the same; however, there are also E and S, where E labels the end of each named entity chunk and S which labels the named entities with only one token.

B. Evaluation Metrics

CoNNL and MUC metrics are considered as standard evaluation metrics for NER tasks. In general, MUC results are higher than CoNNL results due to its considerations on partial scores. Considering the differences between metrics, CoNNL can be called as strict matching metric since it is required to have exact match of boundary and entity type of the ground truth and the predictions. Furthermore, MUC can be called as relaxed matching metric since exact match of boundary and entity types are not required, where partially correct matching is also considered.

1) CoNLL: Computational Natural Language Learning

CoNLL defines the precision and recall for named entities. This metric is considered as "exact-match evaluation" i.e. tagging output must be matched exactly with ground truth in terms of classes or types. We explain precision, recall and f1 as follows.

a) Precision

Precision indicates the percentage of the True Positive (TP), i.e. correctly predicted, named entities over the all words that are predicted as named entity; in other words, all TP and incorrectly predicted False Positive (FP). Thus, it is calculated as in (1).

$$Precision = TP / (TP + FP) \quad (1)$$

b) Recall

Recall indicates the percentage of the TP named entities over all named entities predicted as correctly classified (TP) and missed classified False Negative (FN). Thus, it is calculated as in (2).

$$Recall = TP / (TP + FN) \quad (2)$$

c) F1 Score

In order to achieve a comparison, F1 score is calculated using Precision (PR) and Recall (RE). In addition, F1 score is calculated harmonically instead of a simple average so that extreme values can also be considered. Thus, F1 score is computed as in (3).

$$F1 = 2 * ((PR * RE) / (PR + RE)) \quad (3)$$

2) Message Understanding Conference (MUC)

In MUC, detailed evaluation metrics are presented. This evaluation process is called "relaxed-match evaluation", since there are partial categories of errors: Correct (COR), Incorrect (INC), Partial (PAR), Missing (MIS), and Spurious (SPU). COR and INC refer to correctly and incorrectly matched entities, respectively. PAR means that predicted and truth label are not the same but similar in the matching for boundary or class type. MIS states that truth label is not captured. SPU means that the system predicts something that does not exist in the truth, i.e. labeled as O in ground truth. With these new categories, precision and recall calculations needs to be modified as in (4), (5) and F1 calculation is the same as (3).

$$PR = (COR+0.5*PAR) / (COR+SPU+0.5*PAR) \quad (4)$$

$$RE = (COR+0.5*PAR) / (COR+MIS+0.5*PAR) \quad (5)$$

III. METHODS AND STUDIES IN TURKISH NAMED ENTITY RECOGNITION

Since Turkish is an agglutinative language, it has more information in morphemes. As an example, other agglutinative languages can be Finnish and Czech, called as morphologically rich languages. In Turkish NER studies, employing morphological and syntactic features is commonly observed in order to increase the performance of the model. We divide existing studies according to the technical solutions into five categories: rule-based, statistical learning, hybrid, neural network and transformer-based methods.

A. Rule-Based Methods

Rule-based systems are problem solving tools depending on human expertise. It simply works as giving rules that are defined by human knowledge to the system, so that the system can make decision accordingly. This provides to handle complex problems by combining specific rules about them. Since rule-based system requires human expertise, it is not easy to design. Moreover, the rules extracted for a domain cannot be adaptable for another domain. Although domain adaptation is a common issue for rule-based systems, they can give promising Precision scores for specific domains [5].

Reference [5] presented the first rule-based NER system on News Articles [1], child stories, and historical texts with 78.7%, 69.3% and 55.3% F1 scores, respectively. They did not use the capitalization and punctuation rules in order to make the system robust for noisy texts.

Reference [6] developed an automatic rule learning system and reported F1 score of 91.08% on the TurkIE dataset manually tagged on terrorism using both online and printed newspapers, where the dataset is not publicly available anymore.

Reference [7] implemented a rule-based system on Twitter datasets by adapting its rules to fit the datasets by relaxing capitalization constraint and by diacritics-based expansion, and they also employed a simplistic normalization scheme. They experimented on two different Turkish Twitter datasets. They reported 48.13% F1 score on the first dataset [7] and 38.01% F1 score on the second dataset [3], without capitalization and normalization.

B. Statistical Learning Methods

Statistical learning methods, being framework for machine learning, are briefly function estimation based on data. Hidden Markov Models (HMM) [8] and Conditional Random Field (CRF) [9], which are to model sequential data, are usually used in NER studies as a statistical learning approach.

Reference [1] presented a statistical learning method based on Hidden Markov Models for Turkish NER by achieving 91.56% F1 score on a news domain, News Articles [1].

Reference [10] proposed the first study employs CRF in NER, where several morphological features are used to infer hidden states. They provided improvement over their own baseline by 7.6%, ending up with 88.94% F1 score employing roots and morphological features of words. They used morpheme-level tokenization method which represents the word as root and morphological feature states on News Articles [1].

Reference [11] presented initial explorations on the usage of rich morphological structure as features to the CRF. They proposed a framework including gazetteers, two-level morphological analyzer [12], and morphological disambiguator [13]. They reported the highest results until then in both MUC (95%) and CoNLL (92%) metrics with gazetteers, and CoNLL (89.59%) without gazetteers.

Reference [3] prepared three new informal Turkish dataset whose domains are Twitter, Speech-to-Text Interface, and a Hardware Forum. They used the same method as in [11] with an addition of a normalizer at the morphological processing step in order to normalize noisy text. They created three different models composing of different sets of features and evaluated them with and without normalization. They reported 91.64% F1 score on a formal dataset [1] and relatively lower success rates (~54% and ~19%) on different informal datasets.

C. Hybrid Methods

Hybrid methods are the composition of rule-based methods and statistical learning methods. Reference [16] proposed the first hybrid named entity recognizer. They were the first to propose a rule-based system for Turkish NER [5], and in this study improved it to learn from annotated data when available. By this, they achieve a hybrid system employing the high success rate of rule-based system on the data they used in [5] and re-annotate them.

Reference [17] proposed a hybrid model that makes use of hand-crafted features. Dependency parsing related features together with other features are used to boost NER performance. The model employs CRF and uses News Articles, and the reported F1 score is 89.89%.

D. Neural Network-based Methods

In the perspective of Neural Networks, BiLSTM and perceptron are used for Turkish NER studies. BiLSTM is composed of two LSTMs [18], where one takes input sentence in forward direction, and the other one in backward. Since it is a sequence modelling and with this backward-forward structure, BiLSTM is useful for information extraction and constructing relations neighboring words. A common architecture in Turkish NER studies is BiLSTM-CRF model, where word embbeddings are processed via BiLSTM, and named entities are tagged with CRF layer. We illustrate an example of BiLSTM-CRF architecture in Fig. 1.

Reference [20] implemented a semi-supervised learning approach based on neural networks using the framework of [21], who employed regularized averaged perceptron algorithm. They adopted a fast-unsupervised method to learn continuous vector representations of the words and used them with language independent features. They improved previous state-of-the-art result by 2.26% over [11] (overall 91.85%) and without using gazetteers for Turkish and Czech by 1.53% over [22] (overall 75.61%). Unlike the previous works, their system does not make use of any language dependent features; thus, it is implementable for other morphologically rich languages such as Czech.

Reference [23] obtained word embeddings and used them as features to train Window Approach Network (WAN) of the SENNA, which is a framework proposed in [24]. They trained the word embeddings on a large and merged unannotated text corpus, Boun Web Corpus [13] and Turkish Wikipedia containing around 500M tokens, with a vocabulary of size 954K. Evaluating on six different data sets from previous studies, they improved the F1 score on Twitter Set [7] to 57.2% from 48.13%, and on Speech Data Set [3] to 71.54% from 50.84%.

Reference [25] utilized a semi-supervised learning approach based on neural networks where a regularized averaged multiclass perceptron was used. They employed Skip-gram model to obtain word vectors using word2vec [26] on Boun Web Corpus, together with language independent features that are engineered to work better on informal text types. In addition, for supervised learning steps, they used News Articles [1], Twitter Dataset [3], and Twitter Set [7]. They achieved the state-of-the-art until then for Turkish Twitter Dataset and Twitter Set scores 48.96% and 56.79%, respectively.

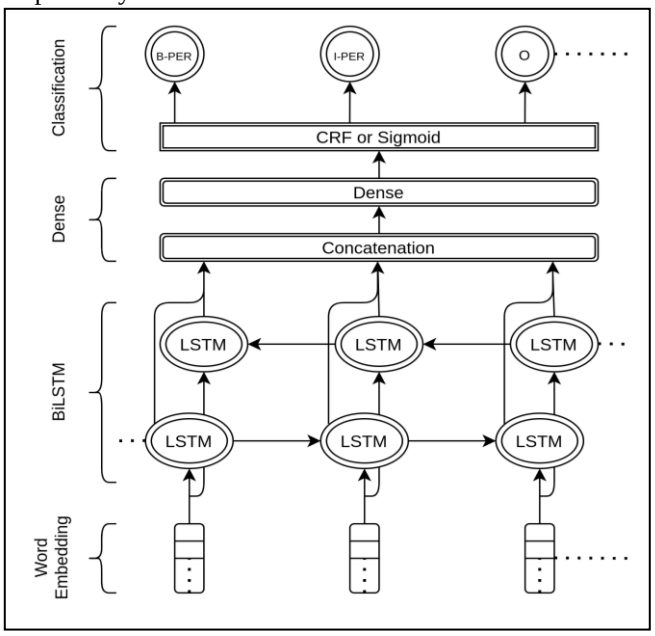


Fig. 1. An illustration of the BiLSTM-CRF model architecture. For a given sentence, word representations of each word are fed into the LSTM cells. The output of LSTMs is concatenated and passed to a dense layer. Final outputs of dense layer are given to a classifier layer, such as CRF or sigmoid to produce predictions for each word.

Reference [19] also utilized a neural network application. They implemented BiLSTM and Deep-BiLSTM, using stacked layer on top of BiLSTM with several layer numbers. They improved the F1 score of [14] by 0.10%, and the best model is reported with 93.69% F1 score for 4 layers of stacked BiLSTM.

Reference [14] employed the RNN [15] structure to create context vector embeddings, and train BiLSTM model with this context vector embeddings to predict named entities.

Reference [4] presented transfer learning by adopting a deep recurrent neural network model without using any hand-crafted features. As input to the BiLSTM-CRF model, different levels of word embeddings are used. A single CRF model is trained on a large dataset which is the re-annotated version of News Articles, and the other one is trained on a small dataset which is noisy-informal Twitter dataset. Thus, the model learns from both data set jointly, and transfer learning achieves 67.39% F1 score on Turkish noisy data.

E. Transformer-Based Methods

Transformer is a model architecture that employ the self-attention algorithm to capture the relations among words and sentences [27]. BERT [28] is a recent pre-trained model that employs Transformer for masked language modeling and next sentence prediction. By using transfer learning, the pre-trained BERT model can be fine-tuned for the task of named entity recognition by using a Turkish dataset. However, there are limited studies that incorporate Transformer-based models for the task of Turkish named entity recognition.

Reference [2] empirically investigated the recently used neural architectures and concluded that transfer-based networks overcome the limitations of BiLSTM networks. They also proposed a Transformer-based network with a CRF layer on top, and reported 95.95% F1 score, which is the current state-of-the-art result on News Articles [1] dataset. We illustrated the architecture of pre-trained transformer-based masked language model with the classification layer of CRF in Fig. 2.

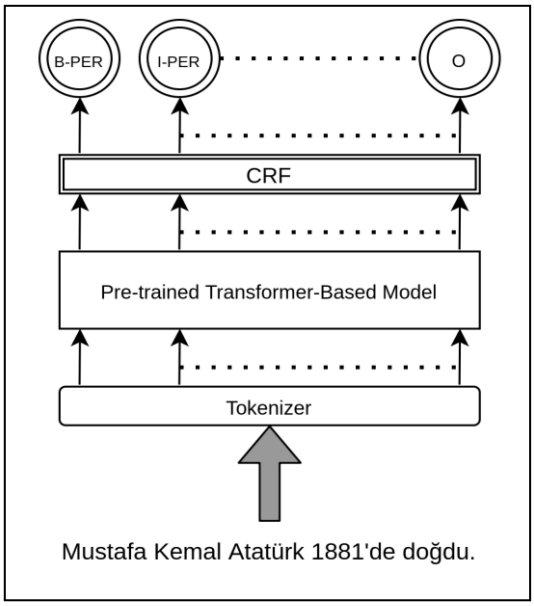


Fig. 2. An illustration of the Transformer-based CRF model architecture. A given sentence is fed into a tokenizer. The tokenizer tokenizes the sentence into smaller sub-units, such as sub-words or characters. Produced tokens are given to the pre-trained Transformer-based model. The output of the Transformer-based model is fed into the CRF layer to produce predictions for each token.

In overall, we summarize the results for different methods on News Articles [1] and Twitter Dataset [3] for Turkish NER in Table II.

IV. DISCUSSION

In most of the studies, datasets are coarse grained in named entity tags, which are PLO or additionally TIMEX and NUMEX. Therefore, a detailed dataset can be prepared with fine-grained named entities, which would be a significant contribution for Turkish NER. As an example of fine-grained labels, entities can be divided further into topics under coarse grained ones: Politician and Influencer for PERSON, City and Country for LOCATION, Governmental and Charity for ORGANIZATON.

There is a lack of publicly available informal and formal datasets in Turkish named entity recognition. In most of the studies, the same dataset is used as benchmark, i.e. News Articles [1] as an instance of formal text, and Twitter Dataset [3] as an instance of informal text. Using the same dataset can provide researchers to compare their results; however, larger datasets with variety of instances and NER tags would be needed to develop robust and diverse models for Turkish NER.

Turkish is a morphologically rich language. Thus, it has a productive nature, which causes generation of many words from a given word root. Therefore, the data sparseness problem in model training can occur in Turkish NER studies. Considering agglutinative nature of Turkish, [6] utilizes morphological features to handle this problem and reports that it is required to use such features. Furthermore, more analyzes can be done via a two-level morphological analyzer [12], which produces the possible morpheme analyzes for each word. The output of analyzer can then be given to a morphological disambiguator [13], in order to get the most probable analyzes for the words. Such techniques are employed in [11], where the output of the analyzer includes both words and morphological features of the words.

TABLE II. TURKISH NER STUDIES REVIEW

Data	Study	Method	F1 (%)
News Articles [1]	[2]	BERTurk-CRF	95.95
	[19]	Deep-BiLSTM	93.69
	[14]	BiLSTM	93.37
	[11]	CRF	91.94
	[1]	HMM	91.56
	[10]	CRF	88.94
Twitter Dataset [3]	[4]	BiLSTM-CRF	67.39
	[25]	Reg. Avg. Perp.	48.96
	[7]	Rule-based	38.01
	[3]	CRF	19.28
	[23]	WAN	15.43

Turkish noisy informal data, e.g. Turkish tweets, have special text segments; such as emoticons, abbreviations, slang words, and repeated words. A text normalization process may be adapted to the model architecture, which can eliminate possible noise [3].

V. CONCLUSION

We provide a brief survey on Turkish Named Entity Recognition. In short, rule-based systems report promising success; however, they have domain adaptation issue. Furthermore, rule-based systems require more human skills and domain expertise to design. Using statistical machine learning approaches give better results by considering their adaptation to other domains. In statistical approaches for Turkish NER, the first study is [1]. Next, CRF started to be used [10] and it is fed by hand-crafted features, syntactic features, language independent features, word embeddings and so on. CRF is proved to be a successful classifier in most of the NER studies due to capture label transition on IOB2 tagging. However, it is still lack of capturing the semantic relations among the words in the sentences such as in Transformer-based networks [27], since CRF is seeking for conditional probabilities considering one previous label. After proposal of word2vec [26], NER scores developed fast. Word embeddings need to be trained with large unlabeled data sets, and most of studies in Turkish NER employ word embeddings in terms of character-level, word-level and so on.

In future work, formal and informal publicly available datasets can be annotated for Turkish NER studies since there is lack of informal and formal publicly available dataset for Turkish. Moreover, a comprehensive comparative study with various well-resulted models under the presented approaches can be tested on different datasets, and results can be analyzed.

REFERENCES

[1] G. Tür, D. Hakkani-Tür, and K. Oflazer, "A statistical information extraction system for Turkish", Natural Language Engineering, vol 9, no 2, pp. 181–210, 2003.

[2] G. Aras, D. Makaroğlu, S. Demir, and A. Cakir, "An evaluation of recent neural sequence tagging models in Turkish named entity recognition", Expert Systems with Applications, vol 182, p. 115049, 2021.

[3] G. Çelikkaya, D. Torunoğlu, and G. Eryiğit, "Named entity recognition on real data: a preliminary investigation for Turkish", in 2013 7th International Conference on Application of Information and Communication Technologies, 2013, pp. 1–5.

[4] E. K. Akkaya and B. Can, "Transfer learning for Turkish named entity recognition on noisy text", Natural Language Engineering, vol 27, no 1, pp. 35–64, 2021.

[5] D. Küçük and Others, "Named entity recognition experiments on Turkish texts", in International Conference on Flexible Query Answering Systems, 2009, pp. 524–535.

[6] S. Tatar and I. Cicekli, "Automatic rule learning exploiting morphological features for named entity recognition in Turkish", Journal of Information Science, vol 37, no 2, pp. 137–151, 2011.

[7] D. Küçük and R. Steinberger, "Experiments to improve named entity recognition on Turkish tweets", arXiv preprint arXiv:1410. 8668, 2014.

[8] L. Rabiner and B. Juang, "An introduction to hidden Markov models", ieee assp magazine, vol 3, no 1, pp. 4–16, 1986.

[9] J. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data", 2001.

[10] R. Yeniterzi, "Exploiting morphology in Turkish named entity recognition system", in Proceedings of the ACL 2011 Student Session, 2011, pp. 105–110.

[11] G. A. Şeker and G. Eryiğit, "Initial explorations on using CRFs for Turkish named entity recognition", in Proceedings of COLING 2012, 2012, pp. 2459–2474.

[12] K. Oflazer, "Two-level description of Turkish morphology", Literary and linguistic computing, vol 9, no 2, pp. 137–148, 1994.

[13] H. Sak, T. Güngör, and M. Saraçlar, "Turkish language resources: Morphological parser, morphological disambiguator and web corpus", in International Conference on Natural Language Processing, 2008, pp. 417–427.

[14] O. Güngör, S. Üsküdarlı, and T. Güngör, "Recurrent neural networks for Turkish named entity recognition", in 2018 26th Signal Processing and Communications Applications Conference (SIU), 2018, pp. 1–4.

[15] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors", nature, vol 323, no 6088, pp. 533–536, 1986.

[16] D. Küçük and A. Yazıcı, "A hybrid named entity recognizer for Turkish", Expert Systems with Applications, vol 39, no 3, pp. 2733–2742, 2012.

[17] A. Akdemir and T. Güngör, "A detailed analysis and improvement of feature-based named entity recognition for turkish", in International Conference on Speech and Computer, 2019, pp. 9–19.

[18] S. Hochreiter and J. Schmidhuber, "Long short-term memory", Neural computation, vol 9, no 8, pp. 1735–1780, 1997.

[19] A. Güneş and A. C. TantuĞ, "Turkish named entity recognition with deep learning", in 2018 26th Signal Processing and Communications Applications Conference (SIU), 2018, pp. 1–4.

[20] H. Demir and A. Özgür, "Improving named entity recognition for morphologically rich languages using word embeddings", in 2014 13th International Conference on Machine Learning and Applications, 2014, pp. 117–122.

[21] L. Ratinov and D. Roth, "Design challenges and misconceptions in named entity recognition", in Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009), 2009, pp. 147–155.

[22] M. Konkol and M. Konopík, "CRF-based Czech named entity recognizer and consolidation of Czech NER research", in International conference on text, speech and dialogue, 2013, pp. 153–160.

[23] K. D. Onal and P. Karagoz, "Named entity recognition from scratch on social media", in Proceedings of 6th International Workshop on Mining Ubiquitous and Social Environments (MUSE), co-located with the ECML PKDD, 2015, vol 104.

[24] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch", Journal of machine learning research, vol 12, no ARTICLE, pp. 2493–2537, 2011.

[25] E. Okur, H. Demir, and A. Özgür, "Named entity recognition on Twitter for Turkish using semi-supervised learning with word embeddings", arXiv preprint arXiv:1810. 08732, 2018.

[26] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space", arXiv preprint arXiv:1301. 3781, 2013.

[27] A. Vaswani et al., "Attention is all you need", in Advances in neural information processing systems, 2017, pp. 5998–6008.

[28] J . Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding", arXiv preprint arXiv:1810. 04805, 2018.

Calculation of Beta Energies of Ground-State Proton and Neutron Matching Gaps with Artificial Neural Network

Muhammed Mustafa ORHAN
Department of Energy Science and Technologys
Sivas Cumhuriyet Universtiy
Sivas, Turkey
m.mustafaorhan97@gmail.com

Serkan AKKOYUN
Department of Physics -Artificial Intelligence Systems and Data Science Application and Research Center,
Sivas Cumhuriyet University
Sivas, Turkey
sakkoyun@cumhuriyet.edu.tr

***Abstract*: Beta decay is a type of decay that occurs as a result of an excess of protons or neutrons in nuclei. This decay in the nucleus occurs when one nucleon turns into another nucleon. But even though this transformation seems like a random process, it has to obey the laws of conservation. In this study, artificial neural network method was used to estimate ground state proton and neutron coupling gap energies. The obtained results were able to successfully make predictions to the expected values, including different artificial neural network models. The obtained data approached the expected values with a very low error rate. Therefore, the results obtained with the help of the artificial neural network structure have shown that machine learning can solve such sensitive and complex systems.**

Keywords: Beha, Beta Decay, Artificial Neural Network, Pairing gap for proton, Pairing gap for neutrons

I. INTRODUCTION

Ionizing radiations are classified as charged particle and uncharged radiations. Charged particle radiation includes various particles such as beta, protons and ions. Uncharged particles include gamma, X-ray photons and neutrons. Beta particles are formed as a result of nuclear reactions. Their energies range from 0.01 to ~1 MeV. Due to their small mass and charge, they scatter easily and lose their energy after each interaction. The range of betas with 1 MeV energy in matter is about 1mm for liquids and solids, and a few meters in air [1].

Beta decay occurs in a nucleus with too many protons or too many neutrons, when one of the protons or neutrons is converted to another. Beta minus decay, one neutron decays into a proton, one into an electron, and an anti-neutrino. In beta plus decay, a proton decays into a neutron, a positron, and a neutrino. Both reactions occur in different regions of the nuclide table as one or the other will bring the product closer to the stability region. These special reactions take place due to the observance of conservation laws [2]. In this study, artificial neural network (ANN) method was used for the estimation of ground state proton and neutron coupling gaps (Δ_{LNp} and Δ_{LNn}) energies [3]. Data for ANN estimates are taken from Atomic Data and Nuclear DataTables [3]. ANN consists of artificial neurons that can perform functional operations in different layers. It produces its own outputs by calculating the analytical functions of the input value with the weight values. ANN does not need a relationship between inputs and outputs. Therefore, it is called the estimator of a nonlinear function. In recent years, ANN has been used in many fields in nuclear physics. Examples of these are the development of nuclear mass systematics [4], obtaining fission barrier heights [5], obtaining nuclear charge radii [6], estimation of beta decay energies [7], approximation of Z boson cross sections [8, 9], gamma determination of beam angular distributions [10] and estimations of radiation efficiencies for electrons in absorbers [11].

II. MATERIAL AND METHOD

Artificial neural networks (ANNs) are very powerful mathematical tool used when standard techniques fail. An ANN is a mathematical model that mimics brain functionality. It consists of several processing units called neurons [12]. In one of the most common types of ANNs, data flows forward from the input layer to the output layer. For this reason, this type of ANN, which is also used in this study, is named as layered feed-forward ANN. Only neurons in different layers are connected by weighted connections. Between these two layers is a hidden layer, which is seen as a black box. The neurons in the input layer receive the data, after the data is processed in the hidden layer, the neurons of the output layer give the result. The number of neurons in the input and output layers is determined according to the problem variables. Also, the number of hidden layer can be changed from 1 to leading deep learning. There are no rules in determining the number of hidden layers and neurons. After many trials for the problem, the number of hidden layers and neurons that give the closest results to the desired values can be considered. ANN consists of two main stages, one is training and the other is testing of training results. The data of the problem is usually divided into two parts, 80% and 20%, and 80% is used for training the ANN and 20% for testing the ANN. The main purpose in education is to determine the weight values of the connections between each neuron in different layers.

In the training phase of this study, Multilayer Perceptron (MP) [12] and Generalized Feed Forward (GFF) [13] ANN model and Levenberg-Marquardt [13,14] algorithm were used. After the weight values that give the best result are determined, the desired values are tried to be produced in the training data with the created network. The error between the outputs produced by the network and the desired outputs is determined by the MSE. The MSE returns the average of the squares of the difference between the desired and neural network output values. It is not enough to see that the network gives successful results on the training data. It should also be determined whether the network can generalize on such data. This is done on the previously allocated 20% dataset. The generated network is applied to the test data and the outputs of

the network are compared with the desired outputs. If the MSE values are below the desired level during the test phase, it can be said that this network is successful in solving the given problem.

In this study, 416 data were used for the energy estimation of ground state proton and neutron coupling vacancies (Δ_{LNp} and Δ_{LNn}). The energies Δ_{LNp} and Δ_{LNn} O, F, Ne, Na, Mg, Al, Si, P, S, and Cl are randomly allocated for training and testing phases. The inputs of the ANN are the proton number, the neutron number and the mass number. The output of the ANN is Δ_{LNp} and Δ_{LNn}. In Multilayer Perceptron and Generalized Feed Forward ANN models, 6 neurons for single hidden layers and 6-4 neurons for double layers were used and the results were compared with each other.

III. RESULTS AND DISCUSSION

In this study, where we tried to obtain the ground state proton and neutron coupling gap energies, the statistical values of the use of different hidden layer and neuron numbers are given in Table I and Table II. As seen in both tables, it is seen that the values are the best in the training and test data for the GFF h=6-4 ANN model.

TABLE I. FOR Δ_{LNp}V MSE, MIN ABS ERROR, MAX. ERROR AND R DATA (UNITS ARE IN MEV)

ANN	ΔLNp									
	MSE		MAE		Min Abs Error		Max Abs Error		r	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
MP h= 6	0,003	0,002	0,045	0,041	5,7E-04	5,1E-04	0,460	0,120	0,985	0,992
GFF h= 6	0,004	0,003	0,047	0,046	6,0E-04	0,001	0,410	0,14	0,985	0,991
MP h=6-4	0,0025	0,001	0,035	0,034	2,0E-05	6 E-06	0,460	0,120	0,991	0,994
GFF h=6-4	0,0024	0,002	0,034	0,037	2,0E-04	2,0E-04	0,420	0,130	0,991	0,994

In the Δ_{LNp} data set, the MSE value was obtained in the worst single-layer GFF, in the h=6 ANN model. In addition, the data for Δ_{LNn} in MP h=6 ANN model were bad. In Δ_{LNp} and Δ_{LNn}, the linear correlation coefficient (r) value for training and test data in the GFF h=6-4 ANN model is 0.995 and above.

TABLE II. FOR Δ_{LNN} MSE, MIN ABS ERROR, MAX. ERROR AND R DATA (UNITS ARE IN MEV)

ANN	ΔLNn									
	MSE		MAE		Min Abs Error		Max Abs Error		r	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
Mp 6	0,008	0,004	0,057	0,051	5,0E-04	9E-05	0,761	0,161	0,990	0,991
Gff 6	0,007	0,003	0,054	0,049	2,0E-04	3E-04	0,806	0,165	0,991	0,992
Mp 6-4	0,005	0,002	0,041	0,04	0,0002	0,001	0,784	0,153	0,993	0,994
Gff 6-4	0,004	0,001	0,035	0,030	5E-05	3E-04	0,746	0,133	0,994	0,996

In Figure.1, the closest values to the theoretical data were obtained among the GFF h=6/4 results. The data obtained from Atomic Data and Nuclear DataTables/[3] MP h=6 The difference obtained in the ANN structure is higher than the other models. In addition, double-layer ANN models gave better results. It is seen that there are small deviations in both tables. It is seen that the most deviations are between 8-11 data. It has been determined that these deviations are at KeV levels and it has been observed that the studs are within acceptable levels.

As a result, in this study, we obtained the energy levels of ground state proton and neutron coupling vacancies using the ANN method. According to the results we obtained, this method was found to be usable.

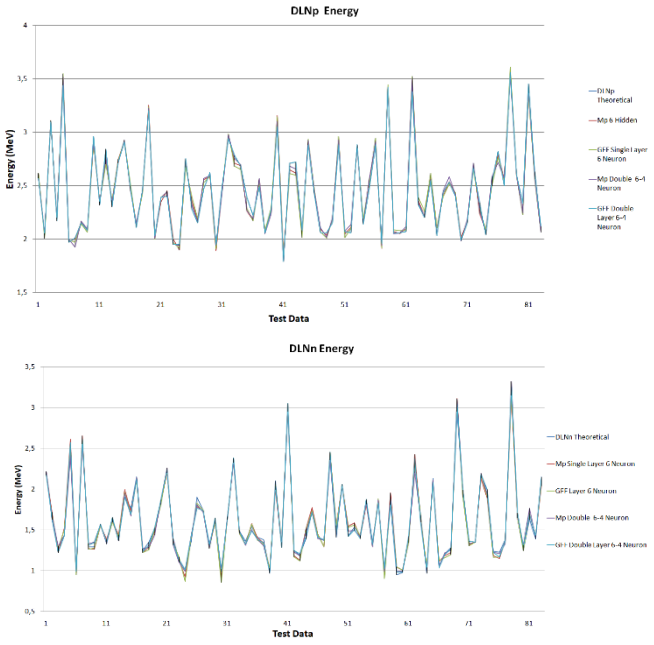


Fig.1 Δ_{LNp} and Δ_{LNn} Energy Outputs graph

REFERENCES

[1] Holbert, K. E., & Murray, R. L. (2015). *Nuclear Energy*. Elsevier Inc.

[2] Guide to the Nuclear Wallchart*. Beta Decay.https://www2.lbl.gov/abc/wallchart/chapters/03/2.html. Date of Access tarihi: 23.11.2021

[3] Möller, P., & Nix, J. R. (1995). Atomic Data Nuclear Data Tables 59 185 Crossref Google Scholar P. Möller, JR Nix, et al. 1997. Atomic Data Nuclear Data Tables, 66, 131.

[4] Akkoyun, S., & Bayram, T. (2014). Estimations of fission barrier heights for Ra, Ac, Rf and Db nuclei by neural networks. *International Journal of Modern Physics E*, *23*(10), 1450064.

[5] Akkoyun, S., Bayram, T., Kara, S. O., & Sinan, A. (2013). An artificial neural network application on nuclear charge radii. *Journal of Physics G: Nuclear and Particle Physics*, *40*(5), 055106

[6] AKKOYUN, S., & Hüseyin, K. A. Y. A. (2020). Estimations of Cross-Sections for Photonuclear Reaction on Calcium Isotopes by Artificial Neural Networks. *Sakarya Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, *24*(5), 1107-1112.

[7] Akkoyun, S., Bayram, T., & Turker, T. (2014). Estimations of beta-decay energies through the nuclidic chart by using neural network. *radiation Physics and Chemistry*, *96*, 186-189.

[8] Akkoyun, S., & Kara, S. O. (2013). An approximation to the cross sections of Z l boson production at CLIC by using neural networks. *Central European Journal of Physics*, *11*(3), 345-349.

[9] Kara, S. O., Akkoyun, S., & Bayram, T. (2014). Probing for leptophilic gauge boson Zl at ILC with by using ANN. *International Journal of Modern Physics A*, *29*(30), 1450171.

[10] YILDIZ, N., AKKOYUN, S., & Hüseyin, K. A. Y. A. (2018). Consistent Empirical Physical Formula Construction for Gamma Ray Angular Distribution Coefficients by Layered Feedforward Neural Network. *Cumhuriyet Science Journal*, *39*(4), 928-933.

[11] Haykin, S. O. (2008). Neural Networks and Learning Machines. 3 Prentice Hall. *New York*.

[12] Abirami, S., & Chitra, P. (2020). Energy-efficient edge based real-time healthcare support system. In *Advances in Computers* (Vol. 117, No. 1, pp. 339-368). Elsevier.

[13] Arulampalam, G., & Bouzerdoum, A. (2003). A generalized feedforward neural network architecture for classification and regression. *Neural networks*, *16*(5-6), 561-568.D.

[14] Marquardt, D. W. (1963). An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, *11*(2), 431-441.

A Substitution Box Generation Method based on Cascading Chaotic Maps

Fırat Artuğer
Department of Computer Engineering
Munzur University
Tunceli, Turkey
firatartuger@munzur.edu.tr

Fatih Özkaynak
Department of Software Engineering
Fırat University
Elazig, Turkey
ozkaynak@firat.edu.tr

***Abstract*—Secure signal processing applications has critical role in many practical application. Researching alternative encryption algorithms is a hot topic, especially to ensure the security of personal data. Substitution box structures have also become increasingly important among these hot research topics since it is one of the basic components in the design of encryption algorithms. In this study, an alternative substitution box design approach is proposed. The proposed algorithm is based on random selections. In the study, a strong entropy source has been generated using cascading chaotic systems. This entropy source is the original aspect of the study. In addition to successfully meeting the substitution box design metrics, it is thought that it can be used for different cryptographic purposes in future studies.**

Keywords—chaos, cryptography, substitution box

I. INTRODUCTION

After the digitalization of data and processes, many regulations have become inevitable. One of the most popular of these regulations is General Data Protection Regulation (GDPR). One of the basic techniques used to meet GDPR requirements is encryption algorithms. In fact, encryption algorithms have been widely used in many applications since data and signal security has become an indispensable part of our lives. For example, 3DES algorithm is used for the security of biometric data in electronic passports [1]. 3DES is an improved derivative of the DES algorithm, one of the block cipher algorithms widely used since the 1970s. Due to the security problems of the DES algorithm, the AES algorithm has been used as a standard after the 2000s [2]. However, side channel attacks on AES algorithm pointed to various weaknesses of mathematically based substitution box (s-box) structures. After demonstrating that random selection based s-box design techniques can be an alternative to side channel attacks [3, 4], many design studies have been proposed [5].

It has been a common approach to realize new designs using chaotic systems in random selection based s-box studies. Researchers aimed to improve s-box design criteria by using different types of chaotic systems. The simple structure of discrete time chaotic systems and the complex nature of continuous time chaotic systems have been the focus of researchers in these different design studies [5]. In this study, an alternative approach to the methods in the literature is proposed. A strong entropy source has been obtained by cascading chaotic systems. Then, 50000 different s-box structures have been generated using this entropy source. These s-box structures have been analyzed under five different s-box design criteria. Among the generated s-box structures, it has been observed that there are two different s-box structures with nonlinearity value is 106.75. This values is the highest nonlinearity value that can be achieved using only chaotic systems. In future studies, it is thought that these successful s-box structures can be used for different information security applications such as block ciphers, image encryption algorithm, key generator and cryptographic mask.

II. PROPOSED METHOD

After chaotic systems produced successful results in many practical applications [6], new chaotic systems have been investigated. It has been shown in the literature that a more robust entropy source can be generated using combinations of chaotic systems [7, 8]. There are various alternatives to analyze chaotic behavior in a system. One of the most popular chaos analysis tools are Lyapunov exponents. This criterion proposed by the Russian mathematician Aleksandr Mikhailovich Lyapunov is expressed as eigenvalues of nonlinear systems, and the positive Lyapunov exponential is used as a quantitative indicator of chaotic behavior [4]. Lyapunov exponents for different chaotic systems are given in Fig. 1 [7]. As can be observed from Fig. 1, it is stated that the entropy of the system, which is the combination of tent and sine map, is higher because the Lyapunov exponent value is larger. The starting point of the study is based on this hybrid structure of chaotic systems.

Four different discrete-time chaotic systems have been used in the study due to their simple mathematical models. Details of the chaotic systems used are given in Eq. (1), Eq. (2), Eq. (3) and Eq. (4) respectively. Another advantage of the selected chaotic systems is that the optimum starting conditions are determined by means of optimization algorithms [9]. In the proposed architecture, four different chaotic systems have been cascaded one after the other, generating a robust entropy source. This entropy source has been transformed into s-box structures by a simple transformation algorithm. The pseudo code of the proposed transformation algorithm is given in Table 1.

$$x_{n+1} = a * x_n * (1 - x_n) \qquad x_n \in [0,1], a \in [3.5, 4] \tag{1}$$

$$x_{n+1} = \begin{cases} b * x_n & x_i < 0.5 \\ b * (1 - x_n) & x_i \geq 0.5 \end{cases} \quad x_n \in [0,1], b \in [1, 2] \tag{2}$$

$$x_{n+1} = c * \sin(\pi x_n) \qquad x_n \in [0,1], c \in [0.85, 4] \tag{3}$$

$$x_{n+1} = x_n + d - \frac{e}{2\pi} \sin(2\pi x_n) \bmod 1 \quad x_n \in [0,1],\ d \in [0, 1], e \in [0, 4\pi] \tag{4}$$

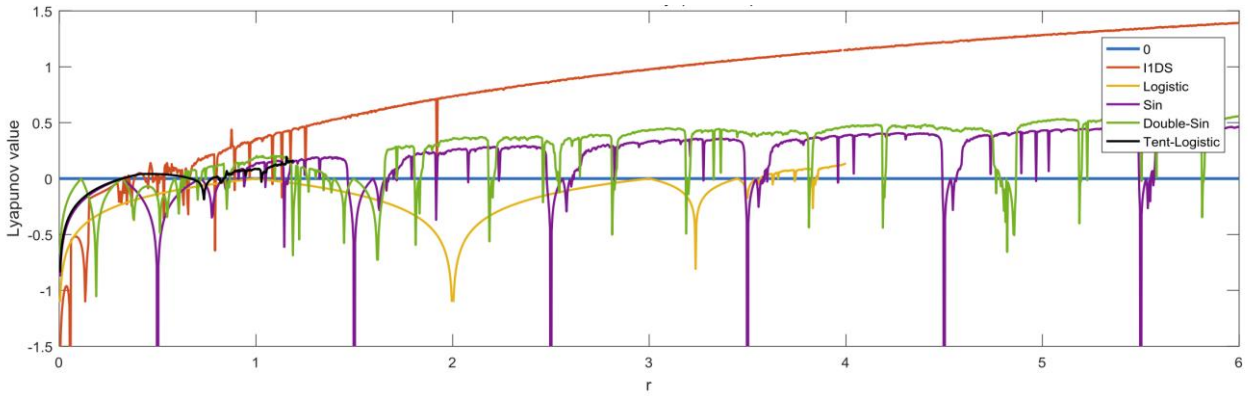


Fig. 1. Comparision of Lyapunov exponents for different chaotic systems

TABLE I. PSEUDO CODE FOR PROPOSED S-BOX GENERATION

```
ChaoticSboxGeneration()
begin
      sbox=[0:255]
      for(k=0; k<256; k++)
      begin
            sbox[k]=-1
      end_for

      i=0;
      control=true
      while(control)
      begin
            value=random(0,1)
            //logistic map
            value=4*value*(1-value)

            //tent map
            b=random(1,2)
            if(value<0.5)
            begin
                  value=b*value
            else
                  value=b*(1-value)
            end_if

            //Sine map
            c=random(0.85, 4)
            value=c*(sin(PI*value))

            //circle map
            d=random(0,1)
            e=random(0,4)
            value=(value+(d-
((e/(2*PI))*sin(2*PI*value))))%1

            value=(value*1000)%256

            if(! Contain(sbox, array))
            begin
                  sbox[i]=value
                  i=i+1
            end_if

            if(i==256)
            begin
                  control=false
            end_if
      end_while
end_procedure

Contain(array, value)
begin
      for(i in len(array))
      begin
            if(array[i]==value) return true
      end_for
      return false
end_procedure
```

The flowchart is given in figure 2 in order to better express the process steps of the algorithm.

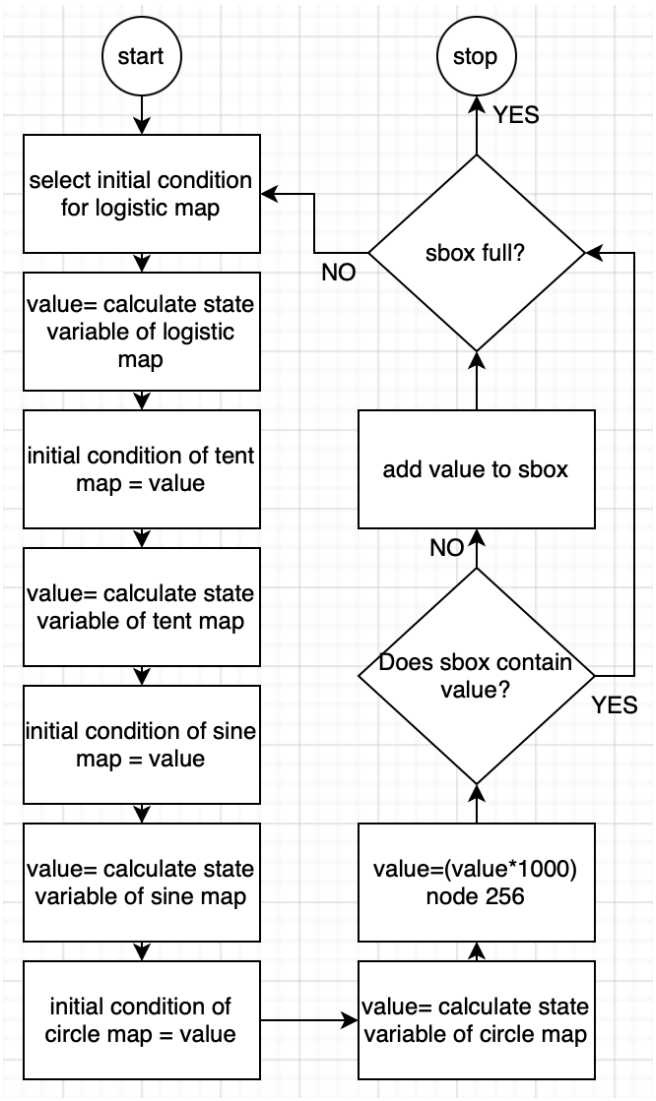


Fig. 2. Flowchart of proposed algorithm

III. ANALYSIS RESULTS

In this study, 50000 different s-box tables have been generated using the method whose details are shared in the Section II. A different initial condition is randomly selected to generate each different s-box structure. There are five basic criteria to test the success of s-box builds. These criteria

analyze how resistant the s-box structure is against statistical, linear and differential attacks. The tests used in the analysis process are Strict Avalanche Criterion (SAC), bijective properties, nonlinearity, Bit Independence Criterion (BIC) and XOR distribution. Ref. [10, 11] can be examined for mathematical explanation of these tests and how they relate to attacks. In addition, free analysis software in Ref. [12-14] can be used to verify the analysis results.

In studies where only chaotic system outputs are converted to s-box values; It has been shown in previous studies that the highest nonlinearity value that can be reached is 106.75 [15]. This result has been reached for two of the 50000 s-box structures produced in this study. These s-box structures are given in table 2 and table 3, respectively. It has been shown that 106.75 can be improved by using different conversions or optimization algorithms. However, since this study is based on using only the raw state of the chaotic outputs, a comparison with the studies based on similar design approach is given in Table 4.

TABLE II. GENERATED SAMPLE S-BOX1

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	148	141	234	176	193	13	227	46	40	97	184	113	138	205	139	91
1	244	48	37	45	154	30	68	228	43	64	59	17	161	83	28	206
2	39	50	8	225	237	105	163	25	136	86	149	9	186	82	174	165
3	254	187	115	18	102	33	209	61	211	60	189	3	29	122	10	63
4	124	169	207	251	129	2	88	27	158	114	65	106	62	173	44	35
5	12	7	222	56	22	162	204	55	210	26	121	168	232	36	79	172
6	217	159	236	201	16	238	142	11	4	23	203	200	90	155	117	110
7	76	51	54	183	94	230	248	119	151	31	195	14	116	166	212	57
8	38	133	32	107	160	34	140	132	24	53	199	80	15	103	58	215
9	127	120	177	41	226	5	208	245	131	42	89	182	255	69	164	249
10	21	0	175	242	87	145	197	99	181	216	6	1	179	147	96	146
11	98	170	224	74	93	150	112	100	137	247	92	108	143	252	47	19
12	95	253	235	75	241	239	250	202	185	180	104	233	192	223	194	188
13	156	152	111	67	178	231	167	77	157	126	153	78	123	49	20	221
14	52	219	101	191	84	72	218	229	73	171	118	70	109	85	134	196
15	130	214	144	125	240	243	220	198	128	213	66	246	135	81	190	71

TABLE III. GENERATED SAMPLE S-BOX2

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	204	221	9	193	34	94	148	2	32	190	241	44	198	145	114	161
1	235	147	227	141	146	38	35	40	62	216	156	164	173	17	210	42
2	19	236	41	21	98	230	64	196	100	22	128	89	229	119	65	4
3	169	226	233	72	219	217	133	130	122	189	39	239	24	54	218	49
4	205	85	252	244	212	11	68	154	0	135	6	55	66	107	63	75
5	47	3	59	159	223	88	33	95	30	58	132	45	224	186	213	253
6	138	25	105	27	109	183	103	222	211	124	36	251	220	153	16	195
7	115	79	185	29	18	101	245	113	46	37	182	171	80	181	168	52
8	127	125	246	87	61	206	199	10	57	152	60	255	247	13	1	31
9	166	200	209	28	250	50	43	23	104	93	14	179	51	15	108	228
10	121	116	70	73	215	191	117	242	243	254	157	180	187	90	123	231
11	197	174	240	139	144	12	225	237	192	207	134	120	151	170	77	118
12	111	178	202	149	131	99	110	155	5	248	184	201	175	69	78	232
13	142	112	86	137	53	208	56	176	214	20	160	67	150	203	7	158
14	172	238	74	194	97	76	26	71	81	48	106	92	177	140	188	162
15	8	136	143	91	163	165	249	82	126	96	129	84	234	102	83	167

The s-box structures, which are the outputs of the proposed generator are compared with 30 different s-box structures. S-box proposals with nonlinearity values below 105 are not listed in Table 4 due to the page limit. Ref. [15] can be examined for other s-box structures with lower nonlinearity value. Analysis results showed that the proposed method outcome in terms of nonlinearity criteria is better than 20 studies. It has the same nonlinearity value as other s-box structures. It has been observed that similar results have been obtained in terms of other s-box design criteria.

TABLE IV. PERFORMANCE COMPARISIONS

S-box	Strict Avalanche Criterion			Nonlinearity			Maximum I/O XOR	Bit Independence Criterion	
	avg	max	min	min	max	avg		SAC	Non.
Ref. [16]	0.5012	0.5156	0.4531	108	112	109	8	0.5012	104
Ref. [33]	0.5017	0.5938	0.4062	106	110	108.5	10	0.4971	104
Ref. [17]	0.5007	0.5175	0.4258	104	110	108	12	0,5006	112
Ref. [41]	0.4988	0.5313	0.4453	106	110	108	12	0.4969	102.85
Ref. [42]	0.4990	0.5781	0.4063	106	110	108	10	0.4961	104.2
Ref. [18]	0.4932	0.5625	0.4932	106	108	107	10	-	102.2
Ref. [15a]	0.4941	0.6094	0.3909	106	108	106.7	10	0.4957	103.5
Ref. [15b]	0.4063	0.4971	0.4063	106	108	106.7	10	0.4994	103.2
Ref. [23]	0.5034	0.6250	0.4219	106	108	106.7	10	0.4951	104
Ref. [34]	0.4976	0.625	0.4062	104	108	106.7	10	0.504	103.5
Table II	**0.4973**	**0.625**	**0.4062**	**104**	**108**	**106.7**	**12**	**0.4985**	**102.36**
Table III	**0.5071**	**0.6094**	**0.42.19**	**104**	**108**	**106.7**	**12**	**0.4976**	**102.93**
Ref. [35]	0.4978	0.5938	0.4375	106	108	106.5	10	0.5003	104.2
Ref. [40]	0.4990	0.5781	0.4063	106	108	106.5	10	0.5033	103.5
Ref. [24]	0.5039	0.5938	0.4219	104	110	106.2	10	0.5023	102.3
Ref. [39]	0.501	0.5781	0.4219	106	108	106.2	10	0.5288	100
Ref. [43]	0.5059	0.5625	0.4375	104	110	106.2	10	0.5026	104
Ref. [32]	0.5012	0.6406	0.4062	104	110	106	10	0.4977	103.5
Ref. [31]	0.5197	0.625	0.4375	104	110	106	10	0.5014	104.2
Ref. [29]	0.5002	0.5938	0.4219	102	108	106	10	0.4968	105.4
Ref. [22]	0.4976	0.5938	0.4219	104	108	105.7	10	0.5032	104
Ref. [9] b	0.4062	0.5938	0.5027	102	108	105.75	10	0.4972	102,36
Ref. [19]	0.5022	0.5781	0.4063	100	110	105.5	32	0.4983	107
Ref. [21]	0.4926	0.5937	0.4062	98	110	105.5	32	0.4994	105.7
Ref. [27]	0.5010	0.6094	0.4063	102	110	105.5	12	0.4988	104.3
Ref. [28]	0.5056	0.5781	0.4375	102	108	105.3	10	0.4971	104
Ref. [20]	0.5059	0.5781	0.4063	102	108	105.2	12	0.5013	104.3
Ref. [37]	0.4987	0.5469	0.4531	104	108	105.25	10	0.4990	102.6
Ref. [36]	0.5037	0.5625	0.4375	102	108	105.25	10	0.4994	102.6
Ref. [9] a	0.5073	0.6094	0.4062	98	108	105.25	10	0.4986	103,86
Ref. [26]	0.5012	0.5938	0.4063	104	106	105	10	0.4994	103.4
Ref. [38]	0.5046	0.6093	0.4750	102	106	105	10	0.5004	103.6

IV. CONCLUSIONS

Alternative cryptographic designs have become increasingly important in order to address the increasing security concerns with the widespread use of cyber-attacks. One of the striking topics in this area is s-box structures. Alternative s-box designs are the focus for both attackers and designers, especially since block cipher algorithms have an important role in their resistance to attacks. In this study, a new s-box design approach is proposed. In the study, it is aimed to increase the irregularity of the entropy source by cascading four different chaotic systems. 50000 different s-box structures have been produced using the proposed method. The success of the generated s-box structures has been evaluated according to five basic s-box design criteria. It has been analyzed that the study is better than many s-box structures with similar design logic in terms of nonlinearity criteria. It is thought that both the entropy source based on s-box design and s-box structures, which are the outputs of the study, can be used successfully in various information security applications.

REFERENCES

[1] H. Orman and P. Streak, "Why Won't Johnny Encrypt ?", IEEE Internet Comput, vol. 19, pp. 90-94, 2015.

[2] J. Daemen and V. Rijmen, "AES proposal: Rijndael," in Proc. 1st Adv. Encryption Conf., CA, USA, 1998, pp. 1–45.

[3] H. Maghrebi, T. Portigliatti, and E. Prouff, "Breaking cryptographic implementations using deep learning techniques," in Security, Privacy, and Applied Cryptography Engineering (Lecture Notes in Computer Science), vol. 10076, C. Carlet, M. Hasan, and V. Saraswat, Eds. Cham, Switzerland: Springer, 2016.

[4] J. W. Bos, C. Hubain, W. Michiels, and P. Teuwen,"Differential computation analysis: Hiding your white-box designs is not enough," in Cryp- tographic Hardware and Embedded Systems (Lecture Notes in Computer Science), vol. 9813, B. Gierlichs and A. Poschmann, Eds. Berlin, Germany: Springer, 2016.

[5] Artuğer, F., & Özkaynak, F. (2021). An effective method to improve nonlinearity value of substitution boxes based on random selection. Information Sciences, 576, 577-588.

[6] S. Strogatz, Nonlinear Dynamics And Chaos: With Applications To Physics, Biology, Chemistry, And Engineering (Studies in Nonlinearity). Boulder, CO, USA: Westview, 2015.

[7] X. Wang, Y. Li, J. Jin, A new one-dimensional chaotic system with applications in image encryption, Chaos, Solitons and Fractals 139 (2020) 110102

[8] M. Alawidaa, A. Samsudina, J. S. Teha, R. S. Alkhawaldehb, A new hybrid digital chaotic system with applications in image encryption, Signal Processing 160 (2019) 45–58

[9] M. Ş. Açikkapi and F. Özkaynak, "A Method to Determine the Most Suitable Initial Conditions of Chaotic Map in Statistical Randomness Applications," in IEEE Access, vol. 9, pp. 1482-1494, 2021, doi: 10.1109/ACCESS.2020.3046470.

[10] T. Cusick and P. Stanica, Cryptographic Boolean Functions and Applica- tions. Amsterdam, The Netherlands: Elsevier, 2009.

[11] C. Wu and D. Feng, Boolean Functions and Their Applications in Cryp- tography. Berlin, Germany: Springer, 2016.

[12] F. Özkaynak, "An analysis and generation toolbox for chaotic substitution boxes: A case study based on chaotic labyrinth rene thomas system," Iranian J. Sci. Technol.-Trans. Elect. Eng., pp. 1–10, 2019. doi: 10.1007/s40998-019-00230-6.

[13] S. Picek, L. Batina, D. Jakobovi, B. Ege, M. Golub (2014) S-box, SET, match: a toolbox for S-box analysis. In: Naccache D, Sauveron D (eds) Information security theory and practice. Securing the internet of things. WISTP 2014. Lecture Notes in Computer Science, vol 8501.

Springer, Berlin, Heidelberg, pp 140–149. https://doi.org/10.1007/978-3-662-43826-8_10

[14] W. A. Stein et al (2013) Sage mathematics software (version 5.10). The Sage Development Team (2013). http://www.sagemath.org. Accessed 27 Jun 2019Analiz programı

[15] Artuğer, F., & Özkaynak, F. (2021). A method for generation of substitution box based on random selection. Egyptian Informatics Journal.doi: https://doi.org/10.1016/j.eij.2021.08.002

[16] D. Lambić, "A novel method of S-box design based on chaotic map and composition method," Chaos, Solitons Fractals, vol. 58, pp. 16–21, Jan. 2014.

[17] X. Zhang, Z. Zhao, and J. Wang, "Chaotic image encryption based on circular substitution box and key stream buffer," Signal Process., Image Commun., vol. 29, no. 8, pp. 902–913, 2014.

[18] B. M. Alshammari, R. Guesmi, T. Guesmi, H. Alsaif, A. Alzamil, Implementing a Symmetric Lightweight Cryptosystem in Highly Constrained IoT Devices by Using a Chaotic S-Box

[19] I. Hussain, T. Shah, H. Mahmood, and M. Gondal, "A projective general linear group based algorithm for the construction of substitution box for block ciphers," Neural Comput. Appl., vol. 22, no. 6, pp. 1085–1093, 2013.

[20] I. Hussain, T. Shah, M. Gondal, and H. Mahmood, "A novel method for designing nonlinear component for block cipher based on TD-ERCS chaotic sequence," Nonlinear Dyn., vol. 73, no. 1, pp. 633–637, 2013.

[21] M. Khan and T. Shah, "A novel image encryption technique based on Hénon chaotic map and S8 symmetric group," Neural Comput. Appl., vol. 25, nos. 7–8, pp. 1717–1722, 2014.

[22] G. Liu, W. Yang, W. Liu, and Y. Dai, "Designing S-boxes based on 3-D four-wing autonomous chaotic system," Nonlinear Dyn., vol. 82, no. 4, pp. 1867–1877, 2015.

[23] D. Lambić, "A novel method of S-box design based on discrete chaotic map," Nonlinear Dyn., vol. 87, no. 4, pp. 2407–2413, 2017.

[24] Ü. Çavuşoglu, A. Zengin, I. Pehlivan, and S. Kaçar, "A novel approach for strong S-Box generation algorithm design based on chaotic scaled Zhongtang system, "Nonlinear Dyn., vol. 87, no. 2, pp. 1081–1094, 2017.

[25] A. Belazi, A. Khan, A. Latif, and S. Belghith, "Efficient cryptosys- tem approaches: S-boxes and permutation–substitution-based encryption, "Nonlinear Dyn., vol. 87, no. 1, pp. 337–361, 2017.

[26] F. Özkaynak, "From biometric data to cryptographic primitives: A new method for generation of substitution boxes," in Proc. ACM Int. Conf. Biomed. Eng. Bioinformat., Bangkok, Thailand, Sep. 2017, pp. 27–33. doi: 10.1145/3143344.3143355.

[27] A. Belazi and A. A. A. El-Latif, "A simple yet efficient S-box method based on chaotic sine map," Optik, vol. 130, pp. 1438–1444, Feb. 2017.

[28] A. Belazi, A. Khan, A. Latif, and S. Belghith, "Efficient cryptosys- tem approaches: S-boxes and permutation–substitution-based encryption," Nonlinear Dyn., vol. 87, no. 1, pp. 337–361, 2017.

[29] F. Islam and G. Liu, "Designing S-box based on 4D-4 wing hyperchaotic system," 3D Res., vol. 8, p. 9, Mar. 2017.

[30] A. Belazi and A. A. A. El-Latif, "A simple yet efficient S-box method based on chaotic sine map," Optik, vol. 130, pp. 1438–1444, Feb. 2017.

[31] X. Wang, A. Akgul, U. Cavusoglu, V. T. Pham, V. H. Duy, and Q. N. Xuan, "A chaotic system with infinite equilibria and its S-box constructing appli- cation," Appl. Sci., vol. 8, no. 11, p. 2132, 2018. doi: 10.3390/app8112132.

[32] U. Çavuşoğlu, S. Kaçar, A. Zengin, and I. Pehlivan, "A novel hybrid encryption algorithm based on chaos and S-AES algorithm," Nonlinear Dyn., vol. 92, no. 4, pp. 1745–1759, 2018. doi: 10.1007/s11071-018-4159-4.

[33] E. A. Solami, M. Ahmad, C. Volos, M. N. Doja, and M. M. S. Beg,"Anew hyperchaotic system-based design for efficient bijective substitution- boxes," Entropy, vol. 20, no. 7, p. 525, 2018. doi: 10.3390/e20070525.

[34] T. Ye and L. Zhimao, "Chaotic S-box: Six-dimensional fractional Lorenz–Duffing chaotic system and O-shaped path scrambling," Nonlin- ear Dyn. vol. 94, no. 3, pp. 2115–2126, 2018. doi: 10.1007/s11071-018- 4478-5.

[35] D. Lambić, "S-box design method based on improved one-dimensional discrete chaotic map," J. Inf. Telecommun., vol. 2, no. 2, pp. 181–191, 2018.

[36] F. Özkaynak, "On the effect of chaotic system in performance character- istics of chaos based S-box designs," Phys. A, Stat. Mech. Appl., vol. 550, Jul. 2020, Art. no. 124072, doi: 10.1016/j.physa.2019.124072.

[37] K. K. Butt, G. Li, F. Masood, S. Khan, A Digital Image Confidentiality Scheme Based on Pseudo-Quantum Chaos and Lucas Sequence, Entropy 2020, 22(11), 1276; https://doi.org/10.3390/e22111276

[38] F. Artuğer and F. Özkaynak,"A novel method for performance improvement of chaos-based substitution boxes," Symmetry, vol. 12, no. 4, p. 571, Apr. 2020.

[39] D. Lambić, A new discrete-space chaotic map based on the multiplication of integer numbers and its application in S-box design. Nonlinear Dyn 100, 699–711 (2020). https://doi.org/10.1007/s11071-020-05503-y

[40] W. Gao, B. Idrees, S. Zafar, and T. Rashid, "Construction of nonlinear component of block cipher by action of modular group PSL(2,Z) on projective line PL(GF(28))," IEEE Access, vol. 8, pp. 136736–136749, 2020, doi: 10.1109/ACCESS.2020.3010615.

[41] M. A. Ben Farah, R. Guesmi, A. Kachouri, M. Samet, "A new design of cryptosystem based on S-box and chaotic permutation", Multimedia Tools and Applications (2020) 79:19129–19150 https://doi.org/10.1007/s11042-020-08718-8

[42] S. Ibrahim, H. Alhumyani, M. Masud, S. S. Alshamrani, O. Cheikhrouhou, G.Muhammad, M. S. Hossain, and A. M. Abbas, "Framework for efficient medical image encryption using dynamic S-boxes and chaotic maps," IEEE Access, vol. 8, pp. 160433–160449, 2020, doi: 10.1109/ACCESS.2020.3020746.

[43] Z. Zhu, Y. Song, W. Zhang, et al. A novel compressive sensing-based framework for image compression-encryption with S-box. Multimed Tools Appl 79, 25497–25533 (2020). https://doi.org/10.1007/s11042-020-09193-x

Calculation of initial speed and stopping distance with ANN

Muhammed Taha ORHAN
Artificial Intelligence and Data Science (Interdisciplinary)
Sivas Cumhuriyet University
Sivas, Turkey
tahaorhan95@gmail.com

Muhammed Mustafa ORHAN
Energy Science and Technology Engineering (Interdisciplinary)
Sivas Cumhuriet University
Sivas, Turkey
m.mustafaorhan97@gmail.com

Abstract: **The stopping distance, is the distance traveled by the driver when he sees a danger, decides to stop the vehicle he is driving and presses the brake pedal until the vehicle comes to a complete stop. This distance is the sum of the distances traveled by the vehicle when the driver made a decision, the brakes were applied and the vehicle slowed down until it came to a stop. In this study, the possible braking time and distance data after the reaction moment were removed from the formula. An artificial neural network was used in the estimation and programming of the data. According to the results obtained, it has been seen that the initial speed and stopping distance can be obtained with this method.**

Keywords: Artificial Neural Network, Stopping Distance, Initial Speed

I. INTRODUCTION (*HEADING 1*)

The stopping distance, is the distance traveled by the driver when he sees a danger, decides to stop the vehicle he is driving and presses the brake pedal until the vehicle comes to a complete stop. This distance is the sum of the distances traveled by the vehicle when the driver made a decision, the brakes were applied and the vehicle slowed down until it came to a stop. The stopping distance s_{stop} is determined using the formula:[1]

$$Sstop = Shp + Shr + Sbrl + Sbr$$

Human detection distance is the distance the driver travels from hazard detection to decision making. Human reaction distance is the distance traveled by the vehicle when deciding to stop the vehicle after the driver realizes the danger. Brake delay distance varies according to the type of brake used in the vehicle.When calculating this distance, it should be noted that the speed must be converted from km/h to m/s. Vehicle braking distance is the distance a vehicle travels from the moment its brakes are fully applied to the moment it stops.Other negligible factors such as rolling resistance or air resistance[2] are not taken into account in these calculations.Calculations related to stopping distance are generally used in traffic accident investigations, crime scene investigations and expert reports.

In this study, artificial neural network (ANN) method was used for stopping distance and initial velocity estimation.ANN consists of artificial neurons that can perform functional operations in different layers.It produces its own outputs by calculating the analytical functions of the input value with the weight values.ANN does not need a relationship between inputs and outputs.Therefore, it is called the estimator of a nonlinear function.ANN is used in many places in the field of transportationEvaluation of artificial neural networks in transportation engineering.[3-4-5-6-7].

II. MATERIAL AD METHOD

Artificial neural networks (ANN) are a very powerful mathematical tool used when standard techniques fail. An artificial neural network is a mathematical model that simulate brain functionality. It basically consists of several processing units called neurons in three different layers[8].In one of the most common types of ANNs, data flows forward from the input layer to the output layer.For this reason, this type of ANN, which is also used in this study, is named as generalized feed forward ANN.only neurons in different layers are connected by weighted connections. Between these two layers is a hidden layer, which is seen as a black box. The neurons in the input layer receive the data, after the data is processed in the hidden layer, the output layer neurons give the result. The number of neurons in the input and output layers is determined according to the problem variables. Also, the number of hidden layer can be changed from 1 to leading deep learning[9].There are no rules for determining the number of hidden layers and neurons. After many trials for the problem, the number of hidden layers and neurons that give the closest results to the desired values can be considered. ANN consists of two main stages; one is training and the other is testing of training results. The data of the problem is usually divided into two parts, 80% and 20%, 80% is used for training the ANN and 20% is used for testing the ANN.The main purpose in training is to determine the weight values of the connections between each neuron in different layers. Levenberg-Marquardt[10,11] algorithm was used in the training phase of this study. After the weight values that give the best result are determined, the desired values are tried to be produced in the training data with the created network. The error between the outputs produced by the network and the desired outputs is determined by the MSE. The MSE returns the average of the squares of the difference between the desired and neural network output values.It is not enough to see that the network gives successful results on the training data. It should also be determined whether the network can generalize on such data. This is done on the previously designated 20% dataset. The generated network is applied to the test data and the outputs of the network are compared with the desired outputs. If the MSE values are below the desired level during the test phase, it can be said that this network is successful in solving the given problem.

In this study, 241 data were used to estimate the initial speed and stopping distance values. The data corresponding to the speed values from 10 km/h to 250 km/h were obtained from the formulas. Braking time, human perception distance, human reaction distance, brake delay distance and brake distance input values, initial speed

and stopping distance values gave output values. In calculations using a single hidden layer, the number of hidden layer neurons 1, 10, 20, 30 were used and the results were compared with each other.

III. RESULTS AND DİSCUSSİON

In this study, which we carried out to obtain the initial velocity and stopping distance estimations, the statistical values of the use of different hidden layer neuron numbers (h) are given in Table I. As can be seen from the table, the higher the number of hidden layers, the higher the MSE and Max. It was seen that the error decreased. The best values are seen to be h=30 for training and testing.

TABLE I. MSE AND MAX. ERROR DATA (IN UNITS (M/S))

GFF Model	V0			
	MSE		r	
	Train	Test	Train	Test
h=1	0,192	0,169	0,9997462	0,9997951
h=10	4E-04	2E-04	0,9999994	0,9999996
h=20	4E-04	3E-04	0,9999994	0,9999995
h=30	4E-04	3E-04	0,9999994	0,9999996
GFF Model	Stopping Distance			
	MSE		r	
	Train	Test	Train	Test
h=1	15,09	7,702	0,9995	0,9997854
h=10	0,020	0,009	0,9999993	0,9999997
h=20	0,019	0,009	0,9999994	0,9999996
h=30	0,018	0,009	0,9999994	0,9999996

The results of the calculations with the lowest and highest number of hidden layer neurons are presented in graphics. Figure 1 shows the ANN estimates based on the training data for the first speed. In this graph, in which the differences between the values in the database we derived from the formula and the values produced by the ANN are plotted, the differences for h=1 are seen on the graph. If h=30, it coincides with V_0.

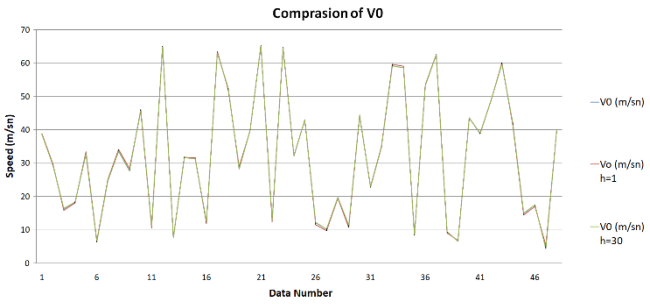


Fig 1. Comprasion of V_0 graphic

Figure 2 shows the ANN estimates based on the training data of the stopping distance. In this graph, in which the differences between the values in the database we derived from the formula and the values produced by the ANN are drawn, it is seen that the difference is more pronounced for h=1. It has been observed that the h=30 ANN structure overlaps with the data derived from the formula.

Fig 2. Comprasion of Stopping Distance graphic

As a result, it was observed that the results improved as the number of neurons increased in ANN. It has been seen that the h= 30 ANN model can be used for estimation of initial speed and stopping distance, braking time, human perception distance, human reaction distance, brake delay distance and brake distance.

IV. REFERENCES

[1] Developing the theoretical model of the effective driving area for providing safe driving and stopping distance in vehicles and testing on real road conditions. Isparta: Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü.

[2] Hartman, J. (2014). Effects of velocity, temperature and rainfall on the friction coefficient of pneumatic tyres and bitumen roads (Doctoral dissertation, RMIT University).I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[3] Faghri, A., & Hua, J. (1992). Evaluation of artificial neural network applications in transportation engineering. Transportation Research Record, 1358, 71.

[4] Collins, A., & Evans, A. (1994). Aircraft noise and residential property values: An artificial neural network approach. Journal of Transport Economics and Policy, 175-197.

[5] Kang, G., Gao, S., Yu, L., & Zhang, D. (2018). Deep architecture for high-speed rail insulator surface flaw detection: Automatic encoder noise removal with multitasking learning. IEEE Procedures for Instrumentation and Measurement, 68(8), 2679-2690.

[6] GÖKTEPE, A. B., AĞAR, E., & LAV, A. H. (2005). Recalculation of mechanical properties in flexible pavements using artificial neural networks. ITU Journal Series D: Engineering, 4(2), 31-42.

[7] KIYILDI, R. K. Estimation of Marshall stability value with artificial neural networks. Niğde Ömer Halisdemir University Journal of Engineering Sciences, 1-1.

[8] Akkoyun, S. Bayram, T. 2021, “Production Cross-Section of 51Cr Radioisotope Using ArtificialNeural Networks” Turk Fen ve Sağlık Dergisi, 2, 133-138. https://dergipark.org.tr/en/pub/tfsd/issue/60096/8282737-K.

[9] Haykin, S., 2008. Neural Networks and Learning Machines . 3. baskı. Hamilton, Ontario, Kanada: Pearson Prentice Hal, s.8-10.

[10] Levenberg, K. (1944). A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics*, *2*(2), 164-168.

[11] Marquardt, D. W. (1963). An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, *11*(2), 431-441.

CSDA Range Estimates of Muon Particle in the Energy Range of 1 GeV to 1 TeV in Some Matters with Artificial Neural Networks

Fatih Enis AYÇİÇEK
Departmant of Energy Science and Technology
Sivas Cumhuriyet University
Sivas, Turkey
Orcid No: 0000-0003-3924-2237

Muhammed Mustafa ORHAN
Departmant of Energy Science and Technology
Sivas Cumhuriyet University
Sivas, Turkey
Orcid No: 0000-0003-1696-7067

Abstract **Muons are different from electrons, which interact with matter in terms of their properties. The high mass of muons creates a resistance against braking and acceleration, so muons can penetrate more effectively in matter despite their coulomb effects. This distance varies according to the properties of the material and the energies of the muons. Based on such data, how far muons can penetrate into a material and the stopping power of the material can be found. Having said that, an artificial neural network can predict how far the muon can penetrate into a material, based on the data and results. With ANN, CSDA distances of different materials at different energy levels were tried to be estimated. In the study, it is seen that this calculation is possible and successfully predictable with machine learning with different ANN models and applications.**

Keywords Muon, CSDA range, artificial neural network

I. INTRODUCTION

Muon is a lepton particle with -1 electric charge and ½ spin. Unlike the electron, it has a larger mass of 100,66 MeV/c2. Such a difference prevents special muons from accelerating and decelerating as much as electrons. These muons can travel longer distances in them[1]. Muons are formed in nature by the decay of pions, released as a result of particle interactions of protons in the content of cosmic radiation (in the atmosphere of the earth) or they can be observed as a result of collisions of proton beams raised to high energies (by human hands)[2]. Although muons have a very short life span of 2.2 microseconds, they can travel long distances due to their speed close to the speed of light[2]. Also, muons cause most of the background radiation on Earth. Like other types of radiation, muons interact with the material and can penetrated a certain distance inside the material. Based on this Distance, we can calculate the materials' average stopping power for muons. The stopping power varies according to the structure of the materials, the energy and type of the incoming particle[1].

In this study, solid elements of muons with different atomic numbers such as B, C, C (Graphite), Al, Si, Ti, Fe, Ag, W, Au, Pb, Rb, Mo, Ta, Hg were used. ANN method was used to estimate the changes in stopping power for muon particles with different energies.[2] ANN is a nonlinear function estimator that can work without any connection between data. For this reason, it is a good tool for physical applications. For example, it is used in many fields such as obtaining nuclear charge radii[3], estimation of beta decay energies[4], estimation of heavy ion fusion reaction cross sections[5].

II. MATERIAL AND METHOD

Artificial neural networks are a very powerful mathematical tool used when standard techniques fail [6]. ANN simulates biological neural networks and creates chain structures. Similar to this structure, ANN basically consists of two stages as input and output. But such simply constructed structures are not sufficient to solve some problems. So it may need to establish more connections. Therefore, the learning and prediction accuracy of the system can be increased by adding hidden layers to the structure. During this learning study, ANN does not need any connection between the data or a linear system. Here, the method is based on determining the weights between neurons that produce the data closest to the desired result. Basically, a learning process consisting of two steps starts with the determination of the weights with the training data. Then, it is tested whether the weights to be used with the test data work correctly. If there are too many discrepancies with the data, the training step is repeated and the weights that produce the closest result are tried to be found. If data with high errors are found, the ANN is reconstructed with a different structure and trained. If the weight values of the transactions can be created well, the neural network starts to produce data with low margin of error.

Multilayer Perceptron(MP), Generalized Feed Forward(GFF) ANN was used in the training phase of the study. Levenberg-Marquardt [7, 8] back propagation algorithm was used for training. Our input data is a total coefficient that includes the material's atomic number (Z), atomic mass (A), density (ρ), energy (T) and coefficients such as ionization, pair formation, braking. Our energy range is determined between 1GeV and 1 TeV. Our output

is one and is the CSDA range. We used two different neural network systems while constructing our neural network, these are feed forward neural networks (GFF) and the other is multilayer neural networks (MP). There is no rule to determine the number of hidden layers or the number of neurons here, it depends on the nature of the problem and the system can be determined after a few trial and error. All of our data is 495. These data are divided into two groups, 420 of them are used for training and the remaining data for testing.

III. Results and Discussion

The total number of data used for the calculations is 495. The number of data used to train the neural network is 420 and for the testing phase 75. First of all, single-layer neural networks used in the creation of neural networks were tested. As a result of the experiments made here, it was seen that the most suitable structure for the problem should include especially 2 hidden layers. The multilayer neural network structure better fits the general outline of the problem. Table 1 and Table 2shows the comparison of the training outputs of multilayer neural networks and feedforward neural networks, and the differences with different hidden layer neuron numbers are shown. When the training data for the MP neural network was examined, trainmse was found in the system with hidden layer neuron numbers of 16-4. Similarly, the network type with the least amount of trainmaxerror is the 16-4 system. Trainmse and trainmaxerror of the system constructed as 5-2 in the training data for the GFF neural network were found to be the best. However, there were differences in the test data. For MP, the lowest test system is 16,-4, and testmaxerror has reached its best value in the 18-6 configuration. The best values of testmse and testmaxerror values for GFF are seen in the figure 8-3.

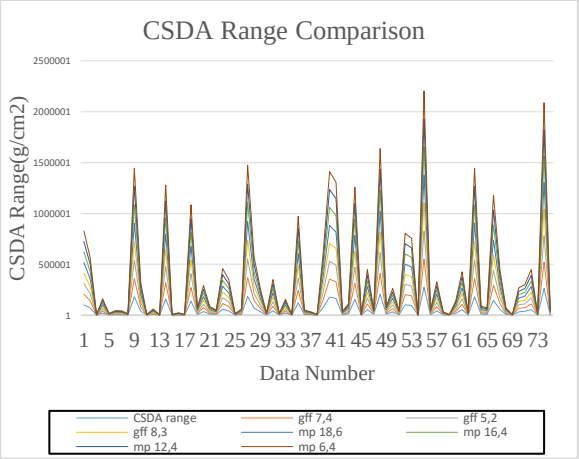
TABLE I. MSE AND R COMPARISON FOR DIFFIRENT ANN

ANN Model	MSE and r			
	r		MSE	
	Train	Test	Train	Test
Mp 6-4	0,999	0,999	3724,233	12302,89
Mp12-4	0,999	0,999	1070,642	13166,6
Mp 16-4	0,999	0,999	407,0103	11617,18
Mp 18-6	0,999	0,999	944,2329	12003,83
Gff 8-3	0,999	0,999	2505,035	13860,41
Gff 5-2	0,999	0,999	1210,745	16449,66
Gff 7-4	0,999	0,999	4931,924	23385,79

TABLE II. MAX AND MIN ERROR COMPARISON

ANN Model	Error Comparison			
	Min Abs Error		Max Abs Error	
	Train	Test	Train	Test
Mp 6-4	0,293	1,102	320,275	650,214
Mp12-4	0,083	0,539	136,237	534,054
Mp 16-4	0,045	0,049	91,394	702,810
Mp 18-6	0,002	0,446	140,388	477,154
Gff 8-3	0,163	0,017	351,555	511,299
Gff 5-2	0,051	0,544	205,329	783,001
Gff 7-4	0,057	1,565	341,071	687,111

Although our MSE values above are high, we can say that it is an acceptable approach since our true error rate is in the range of ±5% for ANNs, which managed to produce close results. The comparison of the test results produced in the chart below with the actual results can be examined.



Comparison of the produced results with the expected value is shown in the graph. Although the results shown in the graph are very close to each other, there is no significant difference between artificial neural networks. This graph is a point stack graph. Therefore, there does not appear to be a difference between the direct series, but to show that ANN produces similar results at the same points. However, if it is necessary to make a comparison between the values found, it is seen that the MP model is somewhat more successful. Among the MP models, more successful predictions were made with a setup with 16-4 hidden layer neurons. In general, it can be said that increasing the number of neurons in the MP model increases the prediction performance. Among the GFF models, the model with the 8-3 configuration is more successful in producing predictions. The GFF model, on the other hand, was able to produce a good prediction with low neuron count compared to MP. However, increasing the

number of neurons in the GFF model does not affect the result linearly.

The results are in agreement with the real data and we can conclude that it is applicable for the use of similar elements.

REFERENCES

[1] Groom, D. E., Mokhov, N. V., & Striganov, S. I. (2001). Muon stopping power and range tables 10 MeV–100 TeV. *Atomic Data and Nuclear Data Tables*, *78*(2), 183-356.

[2] Akerlof, C. W. (2009). Measurement of the Muon Lifetime.

[3] Bayram, T., Akkoyun, S., & Kara, S. O. (2014). A study on ground-state energies of nuclei by using neural networks. *Annals of Nuclear Energy*, *63*, 172-175.

[4] Akkoyun, S., Bayram, T., & Turker, T. (2014). Estimations of beta-decay energies through the nuclidic chart by using neural network. *radiation Physics and Chemistry*, *96*, 186-189.

[6] Hykin, S. (1999). Neural networks: A comprehensive foundation. New Jersey: Printice-Hall.

[5] Akkoyun, S., Bayram, T., Kara, S. O., & Sinan, A. (2013). An artificial neural network application on nuclear charge radii. *Journal of Physics G: Nuclear and Particle Physics*, *40*(5), 055106.

[6] Hykin, S. (1999). Neural networks: A comprehensive foundation. New Jersey: Printice-Hall.

[7] Levenberg, K. (1944). A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics*, *2*(2), 164-168.

[8] Marquardt, D. W. (1963). An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, *11*(2), 431-441.

E-commerce Marketplace Analyzer

Oytun Demirbilek
Inveon Digital Commerce Solutions Limited
İstanbul, Turkey
Oytun.demirbilek@inveon.com

Deniz Köksal
Inveon Digital Commerce Solutions Limited
İstanbul, Turkey
deniz.koksal@inveon.com

***Abstract*— In scope of the great dynamism that the world of e-commerce gained with the post-pandemic changes on customer behavior, it has become very challenging to track and predict the competition standards of the sector and strategize faster. In order to compete in pricing, the seller will need to collect publicly available data about its own and competitor products on the leading marketplaces. Speeding this process up to the rate that the said dynamism occurs is only possible by using the state-of-art crawling and data mining technologies for performance and anonymity. Combined with machine learning for matching accuracy with text recognition and deep learning for forecasting, affective and real time tracking of the industry standards provides an unquestionable advantage to the seller. In this study, automated virtual browser crawlers were developed for high performance scraping of the leading e-commerce marketplaces and the resulting data infrastructure was used for time series forecasting with recurrent neural networks up to an accuracy that allows for agile strategizing within the sector.**

Keywords—E-commerce, web scraping, html crawling, time-series forecasting, dynamic pricing, text recognition.

I. INTRODUCTION

The recent transformation of the retail sector onto digital platforms enabled the manufacturer to not only better know their customers but also their competitors. Discovery, management and correct analysis of the data in accumulation greatly strengthens the industry players' functionality as well as making growth more challenging as competition increases at a faster rate and a higher technology leaning direction.

A global study done in 2017 by KPMG shows that price is the leading sales factor in retail by an average rating of 41% for retail and the main reason that a shopping customer makes an online search during the process is to check for different prices of the same products on different platforms by a mobile search rate of 65%. [1] The leading web-based data source for the same product on different platforms are marketplaces. Equipped with the latest automation blocking capabilities, these platforms do their best to limit scraping size and speed of the publicly available product marketing data for commercial purposes.

Automated and frequent mining, storage, management and exhaustive analysis of this type of data has both technical and industrial challenges. With this paper, we suggest a fool-proof e-commerce marketplace analyzer with the state-of-the-art web scraping technologies with machine learning text recognition and deep learning time-series forecasting abilities.

II. INFRASTRUCTURE

A. Crontab

In order to assign periodic tasks, there is a Linux utility named "crontab". This is used to run commands automatically in the terminal even if there is no connection to the server. Crontab also resolves one more issue in the server: logging. When crontab triggers a terminal command, it can save all outputs printed on terminal to a txt file. Currently, crontab is used to trigger 3 tasks: Running daily crawls, deleting old data in database and running database backup.

B. X-Virtual Frame Buffer

There is an active bot protection tool to block any connection with: headless browsers, no JavaScript, no browsers, IP-cookie mismatch and other false browser fingerprints. Therefore, we need to set up a graphical browser just like a human uses on a computer, but we also need to be able to run it on a headless server. This seems as a great issue at first, but there is a server technique called X-forwarding used by XVFB (X-Virtual Frame Buffer) that sets up a virtual screen, and we can start any display to open a virtual browser on a virtual screen.

C. Chrome & Chromedriver

Our selenium module uses chrome browser and its driver. So we need to be sure that chrome is installed in the system, and the module can access the chromedriver file. It is also crucial that the same version of the Chrome is used. We can check Chrome version in the terminal by typing google-chrome --version and we can download corresponding chromedriver.

III. METHODS

A. Basic Crawlers

A basic crawler is a simple HTML renderer with no browsers and no JavaScript. It simply sends a request to the server, takes the response as HTML and takes information in it. We use beautifulsoup4 in python 3 for HTML rendering and created a fetcher object to identify different types of server responses with HTTP Status Codes. Fetcher object tries to get an acceptable response from the server without any errors. But in the case of redirection (HTTP 301/302) sometimes we accept the new url, sometimes not. Certain marketplaces have a redirection to the category page if we search for the proper keywords. We should accept it since there is non-compact data for search results and we need to use the new URL for crawling. Other examples put a random redirection in the middle of a crawl for a search result. Since it redirects to the mobile page, we cannot accept this.

In the application, we also set up some brand filters to enhance data integrity. Since most of the websites include their filters with JavaScript, there is only one option to include filters in search query: We format the URL string for each marketplace differently. But note this: Some marketplaces include some brands in a different format,. Also there is one more issue we should be aware of: in some marketplaces if the brand does not appear in results, we should not add the brand as a filter.

B. Selenium Crawlers

Selenium crawlers are advanced browser automated scrapers, capable of by-passing the bot protection, rendering JavaScript and collecting the JavaScript rendered data. Uses python3 selenium library to handle browser interaction, requests, JavaScript render and beautifulsoup4 for parsing html. In these types of crawlers, we do not need to handle requests or change URLs to get the intended data.

For bypassing the bot protection crawlers must mimic human behavior, therefore every movement every click should be programmed accordingly. Selenium crawlers take longer time than basic requests crawlers to develop. Selenium crawlers are developed specially for the target site and modified easily if needed.

C. Search Data

In order to provide an easier implementation, we offer a tree-like data structure represented in the figure below:

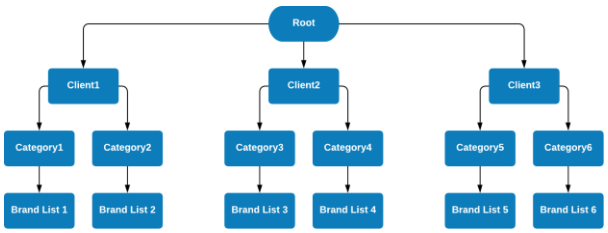


Figure 1: Search Data Tree

There, we simply satisfy all the different requests of categories and brands. Marketplaces are not included in the tree since it makes it more difficult to add/remove categories.

D. Product Matching Algorithm

We followed a product matching task in order to compare prices product wise, which products have been sold with lower or higher prices, and how many. This provides an important insight to the clients, they will assign their own prices much more wisely.

On the other hand, this task is a very challenging one as even for very distinct products such as mobile phones, TVs etc. product names vary greatly across different marketplaces.. We observe other businesses do this task by matching both images and name strings, but in order to keep it simple and implementable, we chose to match with only product names.

a. Jaccard Similarity

At the beginning we implemented a Jaccard similarity algorithm by checking all words and how many of them are common.

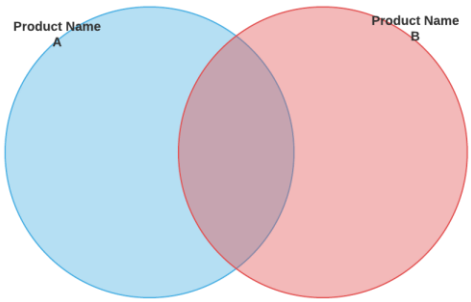


Figure 2: Venn Diagram for Jaccard

We can formulize Jaccard similarity as [2]:

$$J(A,B) = (A \cap B) / (A \cup B)$$

This way handles differences in the word sequence, such as: “Brand A Product 1” is the same as “Product 1 Brand A”. We tried this method, but are not using it right now.

b. Cosine Similarity:

Before we calculate pairwise similarities, we need to change text data into numeric data. It is possible to use Count Vectorizer, Hashing Vectorizer or TF-IDF Vectorizer, these techniques help us to consider words as numbers that the machine can really understand. After we get strings as vectors, then we can calculate the similarity between these 2 vectors.

We chose to use TF-IDF Vectorizer because every word has different weights regarding their uniqueness, other methods cannot provide this information.

It is proven that cosine similarity works well with the TF-IDF method in the literature. Cosine similarity is the cosine value of an angle between 2 vectors in a linear space:

$$\cos\theta = \frac{\vec{a}\cdot\vec{b}}{\|\vec{a}\|\|\vec{b}\|}$$

$$\vec{a}\cdot\vec{b} = \sum_{i=1}^{n} a_i b_i = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$$

$$\|\vec{a}\| = \sqrt{a_1^2 + a_2^2 + a_3^2 + \cdots + a_n^2}$$

$$\|\vec{b}\| = \sqrt{b_1^2 + b_2^2 + b_3^2 + \cdots + b_n^2}$$

c. Term Frequency – Inverse Document Frequency (TF-IDF):

Jaccard and cosine similarity may be seen as very efficient, accurate or adequate at first, but every word has a different importance such as: a serial number is a very unique identifier, but the screen size is not. Therefore, we proposed to use the TF-IDF algorithm to address this issue.

TF-IDF is a well-known and widely used Natural Language Processing (NLP) technique that assigns weights to all words considering their uniqueness. [3] The objective here is to assign higher weights to unique words such as serial numbers, model names etc. After this assignment, we calculate cosine similarity instead of Jaccard similarity because it has previously developed as a high-performance backend in scikit-learn using NumPy.

E. Price Forecasting:

One of the most important issues in the retail market is early-predicting the future prices. Turkey might be a really special case for this, since all the prices usually are going up. However, in such an environment, it is getting more and more crucial to predict possible drops or discounts of a competitor brand in order to be able to take action quickly.

There are various deep learning algorithms to analyze time series data, luckily our tool is collecting very smooth and large data to feed a deep learning model. There are 3 main types of algorithms to find time dependencies in the data: Recurrent Neural Network (RNN), Long-Short Term Memory (LSTM) and Gated Recurrent Unit (GRU).

a. Data Preprocessing:

Preparing a proper data format to feed a time series model is a complicated task. We used the PyTorch library to build everything since it proved its speed and flexibility. We prepared a torch dataset object called: "Ecommerce Dataset" this object inherits from torch's Dataset class and its iteration speed is really fast.

Our dataset object performs 7 main tasks:

1. Read from database: We use Pandas and SQLAlchemy libraries with PyMySQL API to connect our database and retrieve data from one client, one marketplace and one category at a time.
2. Filter out items with inadequate data: We have the data for the past 2 months that means 60-61 time points for each item. There are gaps in the data, or item is removed from the market or item is recently added. In such cases we are looking for only the items with at least 15 time points, and at least 32 products in category.
3. Fill the time gaps: We need to feed the data with a fixed size of time points for every item which is 60 or 61 days.
4. Train-Test Split: There is a much different train-test-split strategy than usual. Since we know all of our products and do not try to predict newly added products, we do not choose the split product-wise, but the time-wise. We split the data like we are in the day just 1 week before and try to predict all of these 7 days. We use past data from that split for training and all data for testing.

   Past-Future Split: This is a functionality that splits the data for a given amount of days. (Going to the past)

   Construct Torch Tensors: Torch tensor objects are automatically back-propagated when we call loss.backward. Therefore, we need to feed the model with torch tensors, also can be converted to any data format such as NumPy. Tensor shapes are very important at each step. When we run into a shape error anywhere, we should observe all the phases of tensor construction by checking its shape by calling sample_tensor.size. The model only accepts the shape of: (batch size, time steps, feature size). Batch size means: number of samples you feed at one forward pass. Time steps means: number of time points you have for one item. Feature size means: number of attributes you feed and get outputs. Now it is 4 as: Price, Discount, Review Count, Rating.

   h_t , C_t : hidden layer vectors.
   x_t : input vector.
   b_f , b_i , b_c , b_o : bias vector.
   W_f , W_i , W_c , W_o : parameter matrices.
   σ , $\tanh$: activation functions.

   Custom Scaling: It is extremely important to normalize your data due to the different scales between your features. This issue will most probably slow your convergence and drop your accuracy drastically. Normally, for classic deep learning tasks, torch has a special layer for normalization before the LSTM layer, it is called LayerNorm, recently introduced in this paper [4] published in 2016. However, there is not any built-in method to denormalize (a.k.a inverse scaling) since we crucially need that, we could not use this method. Sci-kit learn MinMax Scaler did not work also, since we want a forecast for the next 7 days which is very different from the input, MinMax Scaler does not allow that. Instead of all of these, we have built our own scaling object that keeps the main information about the distribution such as mean value and standard deviation, and is able to inverse-scale the data part by part (batch by batch or time by time).

5. There is a special torch object called DataLoader that allows us to iterate through the dataset and split it to random batches. The dataset object returns the past data tensor and the future data tensor if we put into the DataLoader and iterate through. The point of doing that to assign future data as targets and past data as Xtrain.

b. Recurrent Neural Networks (RNN)

RNN algorithm is a very first type of time dependent neuron structure introduced in David Rumelhart's work [5] in 1986. Since it is very fundamental, it can be effective to detect short term time dependencies to produce a proper output.

An RNN Cell can be visualized as in figure below:

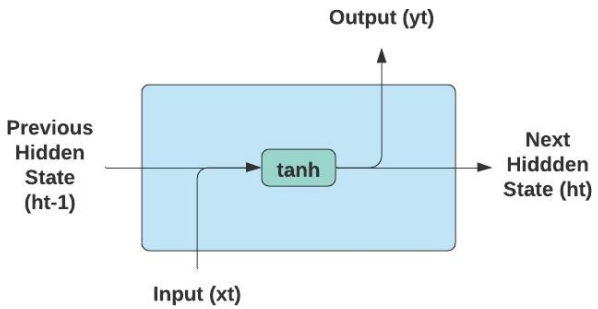


Figure 3:An RNN cell

Note that hidden states (a.k.a hidden layer vectors) represent the learning of the time dependencies, and are needed to be initialized at the beginning of each time sequence.

In our case, RNN considers the latest prices rather than the older ones. Therefore, a long term dependency, such as overall prices, past discounts will be ignored. We observed that there is an accuracy drop up to 45-50% R2 values, and we moved on to train even more advanced models.

c. Long-Short Term Memory (LSTM)

Having tried to implement a Vanilla RNN (a.k.a Basic RNN), we observed that there is not enough prediction accuracy caused by the incapability of learning long term dependencies and switched to LSTM. Introduced by Hochreiter and Schmidhuber in 1997. [6,7] And it is mentioned as revolutionary in many tasks such as speech recognition, handwriting recognition and machine translation especially after 2007, also used in:

- 2015: Google started using an LSTM for speech recognition on Google Voice. According to the official blog post, the new model cut transcription errors by 49%.
- 2016: Google started using an LSTM to suggest messages in the Allo conversation app. In the same year, Google released the Google Neural Machine Translation system for Google Translate which used LSTMs to reduce translation errors by 60%.

An LSTM model greatly sensitive to unnormalized data, we crucially needed to normalize data first. [8] There is a C vector in each LSTM cell called cell state, this represents the importance of that time point. It can affect the model in the long term. The model's strength in detecting different range time dependencies has made it remain as one of the algorithms for time series forecasting long after its discovery. [9]

- An LSTM model works extremely well with the following dropout layer. This layer learns how to randomly exclude certain neurons. We observed that model stability and accuracy greatly increased on testing data when we set the probability parameter as 0.2. This also makes the model avoid overfitting and underfitting conditions. [10]

An LSTM cell can be visualized as:

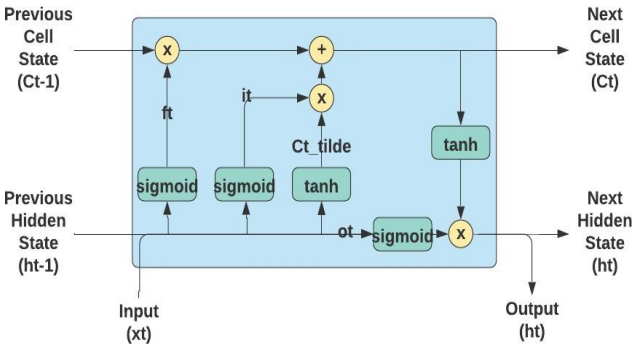


Figure 4: An LSTM cell

And it is formulated as:

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$$

$$h_t = o_t \odot \tanh(C_t)$$

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

In LSTM networks, cell output is just a copy of the next hidden state, therefore, for regression tasks, we should also add a linear layer or its variations at the end. And for the classification tasks, we should add a logsoftmax layer or its variations at the end.

We created LSTM models for each product category and ran various experiments on the data with the LSTM training and testing. we concluded that we can achieve an overall 65%-85% R2 score across all of our product categories

d. Testing Results

Instead of getting very high R2 scores, our purpose was to avoid overfitting and also get proper p-values in order to ensure that our scores did not come by chance. In order to do that, we perform an additional ANOVA Test on predictions and target values. It returns an F-statistic and a P-value. We took the averages across predictions of 7 days.
Results for all categories for a sample brand are as shown in the table below.

Category	R-square	F-statistic	P-value
Cell phone	0.7042964968	18.42544697	0.000226426833
Laptop	0.8602393617	7.175533409	0.06740221779
Desktop	0.654345942	40.47143005	0.1725458616
tablet	0.7053195692	1.197383205	0.3034186499
TV	0.7126599892	0.09456449685	0.7863320354
Camera	0.9701085793	0.06317953842	0.8605825316
Fridge	0.7643109147	1.857466667	0.1803081819
Oven	0.6447670295	0.1157850987	0.7668957016
Stovetop	0.7406147978	0.05561931326	0.8461193138
Iron	0.6879487199	0.1002078446	0.7998118218
Vacuum cleaner	0.7719881962	0.2010222882	0.6998097598
blender	0.8255062668	0.8754612276	0.3535390373
Coffee machine	0.6670498654	9.990889211	0.002602550602
kettle	0.6506197671	0.4997463742	0.5791803458
rondo	0.6194567187	0.04211296197	0.8514346106
mixer	0.7828213703	0.2643120465	0.6676259582
Toast machine	0.793390534	0.2357603665	0.678880698
Air conditioner	0.8624045246	0.01702135872	0.9086484779
epilator	0.8832684138	0.04977553148	0.8437708882
Hair flattener	0.5714813899	0.01387322329	0.9196818384

Table 1: Forecast test results for a sample brand.

F. Multi-Threading

In order to complete our tasks much faster and simultaneously, we run each marketplace crawl for a client in different threads at the same time. This also creates the issue of synchronization of the database write operations. We use python thread queueing technique in order to address this issue. Queue is a data structure that works in first-in-first-out (FIFO) mentality. At initial state, the queue is empty. Then if a crawl finishes, it sends the data to database writing thread and then writing queue. Database writing thread is a different thread that is a child of the main process and checks if the queue is empty or not. After an operation is added, it understands it in 0.1 seconds and starts the operation. While the database thread was in a busy state, other marketplace threads can also add operations to the queue. And the database writing thread waits for the other task to finish, then it takes another operation from the queue.

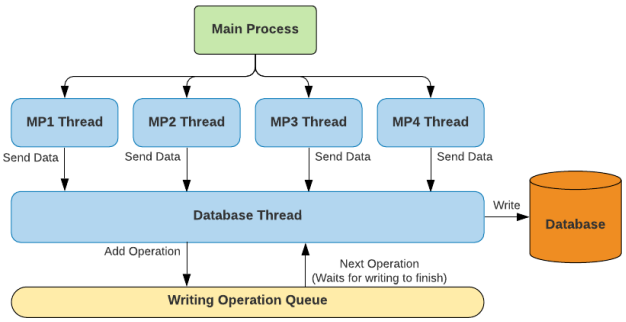


Figure 5: Multi-treading

G. Usage and Reporting

As long as we have large tables, nearly around 15 million rows, we need to summarize this data to much smaller report tables. Because, if we connect this table to Google Data Studio directly, there will be huge SQL queries and the reportwill never be loaded. Therefore, we can see different reports automatically updated from dp_ops.py, individual to each client. If the client requests a matching report, then we will also see the corresponding matching report table, also updated automatically.

The other connector we need is Extract Data, we should use this connector in order to provide incredibly fast-loading reports. It basically connects a different data source that we previously added, and saves the data in its backend. So weno longer need Google Data Studio to execute very complex SQL queries in each load. Also this connector can be scheduled to extract data day by day. However, it has a main drawback: it can only save data up to 10.000 rows. Our reports havethe number of rows at most 5.000 for now, so there is no impact of this drawbackat the moment.

REFERENCES

[1] KPMG, 'Future of Retail', https://assets.kpmg/content/dam/kpmg/xx/pdf/2021/01/future-of-retail.pdf, 2021.

[2] Moulton R, Jiang Y (2018). "Maximally Consistent Sampling and the Jaccard Index of Probability Distributions". *International Conference on Data Mining, Workshop on High Dimensional Data Mining*: 347–356. arXiv:1809.04052. doi:10.1109/ICDM.2018.00050. I SBN 978-1-5386-9159-5.

[3] Robertson, S. (2004). "Understanding inverse document frequency: On theoretical arguments for IDF". *Journal of Documentation*. **60** (5): 503–520.

[4] JL Ba, JR Kiros, GE Hinton, 'Layer Normalization' - arXiv preprint arXiv:1607.06450, 2016.

[5] Rumelhart, D., Hinton, G. & Williams, R. Learning representations by back-propagating errors. *Nature* **323,** 533–536 (1986). https://doi.org/10.1038/323533a0

[6] Schmidhuber, Jürgen (January 2015). "Deep Learning in Neural Networks: An Overview". Neura Networks.61:85:117. arXiv:1404.7828. doi:10.1016/j.neunet.2014.09.003.PMID2546 2637.

[7] Hochreiter, Sepp & Schmidhuber, Jürgen. (1997). Long Short-term Memory. Neural computation. 9. 1735-80. 10.1162/neco.1997.9.8.1735.

[8] Jirsik, T., Trčka, Š., & Celeda, P. (2019, April). Quality of Service Forecasting with LSTM Neural Network. In *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)* (pp. 251-260). IEEE.

[9] LSTM Model Optimization on Stock Price Forecasting, Yifeng Wang, Yuying Liu, 2018 17th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES), 18346178

[10] Dupond, Samuel (2019). "A thorough review on the current advance of neural network structures". Annual Reviews in Control. 14: 200–230.

Interaction Of Nd-Yag Laser Beams With Different Wavelengths With Cadmium Using Artificial Neural Network Method

Ahmet Murad ABED
Department of Physics
Sivas Cumhuriet University
Sivas, Turkey
ahmem1417@gmail.com

Muhammed Mustafa ORHAN Energy
Science and Technology Engineering
(Interdisciplinary)
Sivas Cumhuriet University
Sivas, Turkey
m.mustafaorhan97@gmail.com

SADEQ BADR NOORI
Department of Physics
Tikrit University
Saladin, Iraq
sadiq.b.nouri@st.tu.edu.iq

Abstract: **Cadmium resists corrosion and is used as a protective plate; It is insoluble in water and not flammable. The interaction of the photon with the substance is basically an absorption and scattering reaction. The absorption and absorption of the lasers sent on the cadmium material using the Nd-YAG laser device with the cadmium element in the vacuum environment has been experimentally investigated. However, an artificial neural network can predict how much laser beams will be absorbed or scattered in a material based on the data and results. It has been tried to predict the scattering and absorption of laser beams in cadmium material at different wavelength levels with ANN. In the study, it is seen that this calculation is possible and successfully predictable with machine learning with different ANN models and applications.**

Keywords: Cadmium, Nd-Yag Lazer, Scattering, Absorbing, ANN.

I. INTRODUCTION

Cadmium in its pure form is a shiny silvery metal, whereas Cadmium oxide is a colorless amorphous powder or red-brown crystal. Most of the mined celestine. Chemically it is similar to zinc and mercury. The atomic weight of cadmium results from the mixture of eight stable isotopes. It is a post-transition metal and has two electrons in the s orbital and a full d orbital Cd is corrosion resistant and used as a shielding plate; It is insoluble in water and not flammable. Cd burns in air to form cadmium oxide. Hydrochloric, sulfuric, and nitric acids dissolve cadmium, forming cadmium chloride, cadmium sulfate, and cadmium nitrate, respectively. Cadmium is also used in control rods for nuclear reactors and acts as a neutron poison to control neutron flux in nuclear fission[1] .

Laser is the abbreviation of 'Light Amplification by Stimulated Emission of Radiation'. It can be translated as light amplification of stimulated radiation emission. The first laser beam was produced by Ö,möam in 1960 with a ruby rod[2]. Nd:YAG lasers are optically pumped using a flash lamp or laser diodes. They are one of the most common types of lasers and are used for many different applications. Nd:YAG lasers typically emit light at a wavelength of 1064 nm in the infrared. Nd:YAG absorbs most in the bands between 730-760 nm and 790-820 nm[3]. Krypton flashlights with high output in these bands are therefore more efficient for pumping Nd:YAG lasers than xenon lamps, which produce more white light and therefore consume more energy [4]. The amount of neodymium additive in the material varies according to its use. Doping for continuous wave output is significantly lower than for pulsed lasers. Lightly doped CW bars are optically distinguishable as less coloured, almost white, higher doped bars pink-purple.

In this study, the experimental results of the absorption and transmittance of the cadmium element according to the wavelength of the sent laser beams were calculated by artificial neural network method (ANN). ANN consists of artificial neurons that can perform functional operations in different layers. It produces its own outputs by calculating the analytical functions of the input value with the weight values. ANN does not need a relationship between inputs and outputs. Therefore, it is called the estimator of a nonlinear function. In recent years, ANN has been used in many fields in nuclear physics. Examples of these are the development of nuclear mass systematics[5], obtaining fission barrier heights[6], obtaining nuclear charge radii[7], estimation of beta decay energies[8], approximation of Z boson cross sections[9-10], gamma determination of beam angular distributions[11] and estimations of radiation efficiencies for electrons in absorbers[12].

II. MATERIAL AND METHOD

Artificial neural networks (ANNs) are a very powerful mathematical tool used when standard techniques fail. An artificial neural network is a mathematical model that mimics brain functionality. It basically consists of several processing units called neurons in three different layers [13]. In one of the most common types of ANNs, data flows forward from the input layer to the output layer. For this reason, this type of ANN, which is also used in this study, is named as layered feed-forward ANN. Only neurons in different layers are connected by weighted connections. Between these two layers is a hidden layer, which is seen as a black box. The neurons in the input layer receive the data, after the data is processed in the hidden layer, the output layer neurons give the result. The number of neurons in the input and output layers is determined according to the problem variables. Also, the number of hidden layers can be changed from 1 to leading deep learning [12]. There are no rules for determining the number of hidden layers and neurons. After many trials for the problem, the number of hidden layers and neurons that give the closest results to the desired values can be considered. After the weight values that give the best result are determined, the desired values are tried to be produced in the training data with the created network. The error between

the outputs produced by the network and the desired outputs is determined by the MSE. The MSE returns the average of the squares of the difference between the desired and neural network output values. It is not enough to see that the network gives successful results on the training data. It should also be determined whether the network can generalize on such data. This is done on the previously allocated 20% dataset. The generated network is applied to the test data and the outputs of the network are compared with the desired outputs. If the MSE values are below the desired level during the test phase, it can be said that this network is successful in solving the given problem.

In this study, 701 data on the absorption and scattering reactions of laser beams sent on the cadmium material were used. The wavelength is taken from the experimental results from 400 nm to 1100 nm. The data in the dataset are randomly allocated for the training and testing phases. The inputs of the ANN are the wavelength. It is taken as the output as laser beams that it absorbs and scatters. In calculations using a single hidden layer, 10, 18 and 25 are used as the number of hidden layer neurons and the results are compared with each other.

III. RESULTS AND DISCUSSION

In this study, which we carried out to obtain the scattering and absorption data of laser beams sent to the cadmium material, the ANN data of the absorption and Scattering states, and the statistical values of the use of different hidden layer neuron numbers (h) are given in Table 1. As can be seen from the table, MSE and Max. It was seen that the error decreased. It is seen that the best values for training and testing are h=25.

TABLE I. TRAINING AND TEST ANALYSIS OF ABSORPTION AND SCATTERING DATA.

ANN Type	Absorption Value			
	MSE		r	
	Train	Test	Train	Test
Mp h=10	1,91 e-06	5,71 E-06	0,999993	0,992
Mp h=18	9,1 E-07	3,89 E-07	0,999996	0,996
Mp h=25	9,40 E-07	2,84 E-06	0,999996	0,997
ANN Type	Scattering Value			
	MSE		r	
	Train	Test	Train	Test
Mp h=10	4,32 E-07	3,25 E-05	0,999995	0,997
Mp h=18	4,14 E-07	1,29 E-05	0,999995	0,997
Mp h=25	9,40 E-07	1,68 E-05	0,999996	0,997

The results of the calculations with the number of hidden layer neurons are presented in graphics. Figure 1 shows the ANN estimates based on the data of the scattered laser beams. In these graphs, in which the differences between the values we obtained from the experimental set and the values produced by the ANN were drawn, the most difference was found in the experimental data set in the h= 10 graph. The closest values gave better results at h=18 and the training were more successful for h=30.However, contrary to the result that the values increase with the increase in the number of h seen in Table 1, h=18 gave the best result, and the h=25 ANN model gave results with less deviation than h=10.

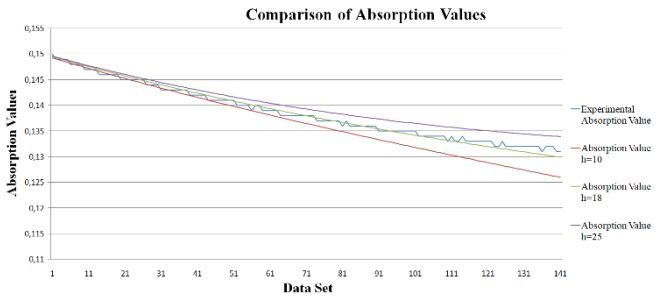


Fig 1. Comparison of absorption values graphic

Figure 2 shows the ANN estimates based on the data of the scattered laser beams. In these graphs, in which the differences between the values obtained from the experimental set and the values produced by the ANN are drawn, the greatest difference was found in the h= 10 data set. The closest values were h=18, and the next best results were h=25. The training is more successful for h=25. However, contrary to the result that the values increase with the increase in the number of h seen in Table 1, h=18 gave the best result, and the h=25 ANN model gave results close to the experimental values with more deviations compared to h=18. As a result, it was observed that the results of 18 neurons in ANN were better. It has been seen that the h= 18 ANN model can be used for calculations at the desired level of absorption and scattering reactions.

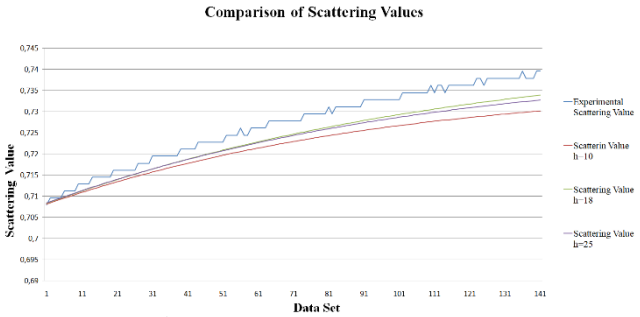


Fig 2. Comparison of Scattering values graphic

REFERENCES

[1] Genchi, G., Carocci, A., Lauria, G., Sinicropi, M. S., & Catalano, A. (2020). Nickel: Human health and environmental toxicology. *International journal of environmental research and public health*, *17*(3), 679.

[2] Maiman, T. H. (2010). 1960. Stimulated optical radiation in ruby. In *A Century of Nature* (pp. 113-114). University of Chicago Press.

[3] Koechner, W. (1988). Damage of optical elements. In *Solid-State Laser Engineering* (pp. 540-558). Springer, Berlin, Heidelberg.

[4] Koechner, W. (2013). *Solid-state laser engineering* (Vol. 1). Springer.

[5] Akkoyun, S., & Bayram, T. (2014). Estimations of fission barrier heights for Ra, Ac, Rf and Db nuclei by neural networks. *International Journal of Modern Physics E*, *23*(10), 1450064.

[6] Akkoyun, S., Bayram, T., Kara, S. O., & Sinan, A. (2013). An artificial neural network application on nuclear charge radii. *Journal of Physics G: Nuclear and Particle Physics*, *40*(5), 055106.

[7] AKKOYUN, S., & Kaya, H. (2020). Estimations of Cross-Sections for Photonuclear Reaction on Calcium Isotopes by Artificial Neural Networks. *Sakarya Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, *24*(5), 1107-1112.

[8] Akkoyun, S., Bayram, T., & Turker, T. (2014). Estimations of beta-decay energies through the nuclidic chart by using neural network. *radiation Physics and Chemistry*, *96*, 186-189.

[9] Akkoyun, S., & Kara, S. O. (2013). An approximation to the cross sections of Z l boson production at CLIC by using neural networks. *Central European Journal of Physics*, *11*(3), 345-349.

[10] Kara, S. O., Akkoyun, S., & Bayram, T. (2014). Probing for leptophilic gauge boson Zl at ILC with by using ANN. *International Journal of Modern Physics A*, *29*(30), 1450171.

[11] YILDIZ, N., AKKOYUN, S., & Hüseyin, K. A. Y. A. (2018). Consistent Empirical Physical Formula Construction for Gamma Ray Angular Distribution Coefficients by Layered Feedforward Neural Network. *Cumhuriyet Science Journal*, *39*(4), 928-933.

[12] Haykin, S., 2008. *Neural Networks and Learning Machines* . 3. Edition. Hamilton, Ontario, Kanada: Pearson Prentice Hal, s.8-10.

[13] Akkoyun, S., & Bayram, T. (2021). Production Cross-Section of 51Cr Radioisotope Using Artificial Neural Networks. *Turkish Journal of Science and Health*, *2*(1), 133-138.