Düzce University Journal of Science & Technology

Research Article

Extreme Learning Machine Algorithms for Prediction of Positive Rate in Covid-19: A Comparative Study

Salih Berkan AYDEMİR [a,*], Funda KUTLU ONAY [a]

[a] *Bilgisayar Mühendisliği Bölümü, Mühendislik Fakültesi, Amasya Üniversitesi, Amasya, TÜRKİYE*
* *Sorumlu yazarın e-posta adresi: salih.aydemir@amasya.edu.tr*

DOI: 10.29130.dubited.999953

Abstract

Various pandemics have been recorded in world history until today. The Covid-19 outbreak, which emerged at the end of 2019, has recently been a hot topic in the literature. In this work, extreme learning algorithms are presented as a comparative study for predicting the positive rate for the countries: India, Turkey, Italy, USA and UK. The features to be used in the learning phase are determined with the F-test feature selection method. For each extreme learning approach, results are obtained for each country with the root mean square error evaluation criteria. Accordingly, the radial basis kernel function produces the best estimation results, while the linear kernel function has the highest RMSE. Accordingly, the lowest RMSE value has been obtained for India as 4.17E-03 with the radial basis kernel function based ELM. Also, since Turkey's data contains too many outliers, it has the highest RMSE value (0.015 - 0.035) in linear kernel method among the countries.

Keywords: *Extreme Learning Machines, Covid-19, Prediction, Feature Selection*

Covid-19'da Pozitif Vaka Oranı Tahmini İçin Aşırı Öğrenme Makinesi Algoritmaları: Karşılaştırmalı Bir Çalışma

Öz

Dünya tarihinde bugüne kadar çeşitli pandemiler meydana gelmiştir. 2019 yılının sonunda ortaya çıkan Covid-19 salgını son zamanlarda literatürde güncel bir konu haline geldi. Bu çalışmada, aşırı öğrenme algoritmaları, en fazla pozitif vakaların görüldüğü ülkeler olan Hindistan, Türkiye, İtalya, ABD ve İngiltere için pozitif oranı tahmin etmeye yönelik karşılaştırmalı bir çalışma olarak sunulmaktadır. F-testi öznitelik seçme yöntemi ile öğrenme aşamasında kullanılacak öznitelikler belirlenir. Her bir aşırı öğrenme yaklaşımı ve her bir ülke için hata ortalama karekökü değerlendirme kriterleri ile sonuçlar elde edilir. Buna göre, radyal tabanlı çekirdek fonksiyonu en iyi tahmin sonuçlarını üretirken, doğrusal çekirdek fonksiyonu en yüksek RMSE'ye sahiptir. Buna göre Hindistan için en düşük RMSE değeri radyal tabanlı çekirdek fonksiyonu tabanlı ELM ile 4.17E-03 olarak elde edilmiştir. Ayrıca Türkiye verileri çok fazla aykırı değer içerdiğinden doğrusal çekirdek yönteminde ülkeler arasında en yüksek RMSE değerine (0.015 - 0.035) sahiptir.

Anahtar Kelimeler: *Aşırı Öğrenme Makinaları, Covid-19, Tahmin, Öznitelik Seçimi*

Geliş: 23/09/2021, Düzeltme: 12/01/2022, Kabul: 30/03/2022

I. INTRODUCTION

Until today, many epidemic diseases have been fought in the history of the world and therefore many people have lost their lives. The World Health Organization (WHO) has developed a number of combat methods in the event of these epidemics and aimed to eliminate or reduce the effects of the disease with these methods. Covid-19 (Coronavirus), which first appeared in Wuhan Province of China in December 2019, has spread exponentially all over the world in a very short time and has become a major threat for global health [1]. The pandemic new type of Coronavirus, which currently does not have an effective treatment and vaccine, has spread to more than 200 countries, infected more than 103 million people, and caused 2,248,089 deaths by February 02, 2021 [2]. Various parameters play an important role in number of people being infected or being affected by the virus. Since, the number of cases, mortality estimates and other calculations related to the Covid-19 virus provide critical information, so studies in this area have become important. Among these studies, there are those carried out using regional data as well as data from all over the world.

There are recent studies on statistical and time-series modelling to model and predict the incidence, prevalence, recurrence rate, mortality rate, spread rate, active and new cases of the pandemic. Li et. al. [3] presented the data-driven analysis approach to estimate the size of the outbreak in other Hubei and other regions of mainland China. Fanelli and Piazza [4] proposed the recursive relations and non-linear fitting approaches to analyze the temporal dynamics of the coronavirus outbreak in China, Italy and France. Wei et. al. [5] performed the global expansion of the epidemic with dynamic map visualization and predicted the spread of the virus across all countries and continents. Ceylan [6] implemented the auto-regressive integrated moving average (ARIMA) model to estimate the prevalence of Covid-19 in Italy, Spain and France. Again, as a study which uses ARIMA, Lukman et. al. [7] investigated the spread frequency of African countries most affected by the epidemic. Almeshal et. al. [8] used deterministic and stochastic modelling approaches to estimate the extent of the spread in Kuwait and to determine the end of the outbreak. Ogundokun et. al. [9] aimed to predict the prevalence of Covid-19 in Nigeria using a linear regression model. Djeddou et al.[10] used four different artificial neural network models to model newly confirmed cases in Algeria. Achterberg et. al. [11] analysed a diverse set of Covid-19 forecast algorithms, including several modifications of NIPA. He et. al. [12] carried out meta-analysis and sensitivity analysis for prediction of the basic reproduction number, the average incubation time, asymptomatic rate infection and the case fatality rate for Covid-19. Hasan et. al. [13] investigated the applicability of the ELM technique to predict the certain Covid-19 cases. Pinter et. al. [14] estimated Covid-19 mortality rate in Hungary using ANFIS and MLP-ICA based hybrid model. Yadav et. al. [15] aimed to predict positive cases, active cases, recoveries, transmission rate and to model the relationship between weather conditions and coronavirus. Rath et. al. [16] proposed a multiple linear regression model for predicting active cases in Covid-19 daily data. Ghosal et al. [17] used linear regression analysis to estimate the number of deaths in India from SARS-CoV-2. Finally, Sujath et. al. [18] predicted the spread of Covid-19 in India using linear regression, multilayer perceptron and vector auto-regression methods.

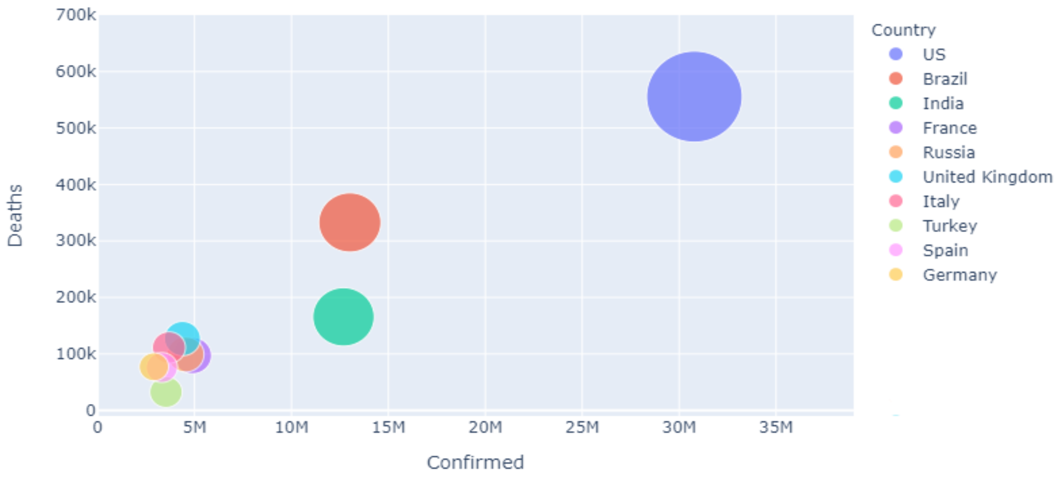
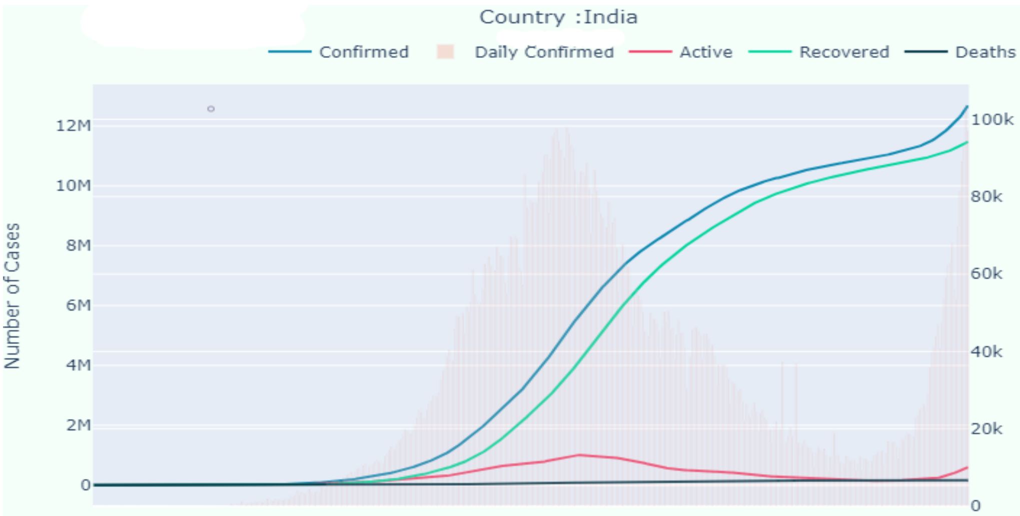
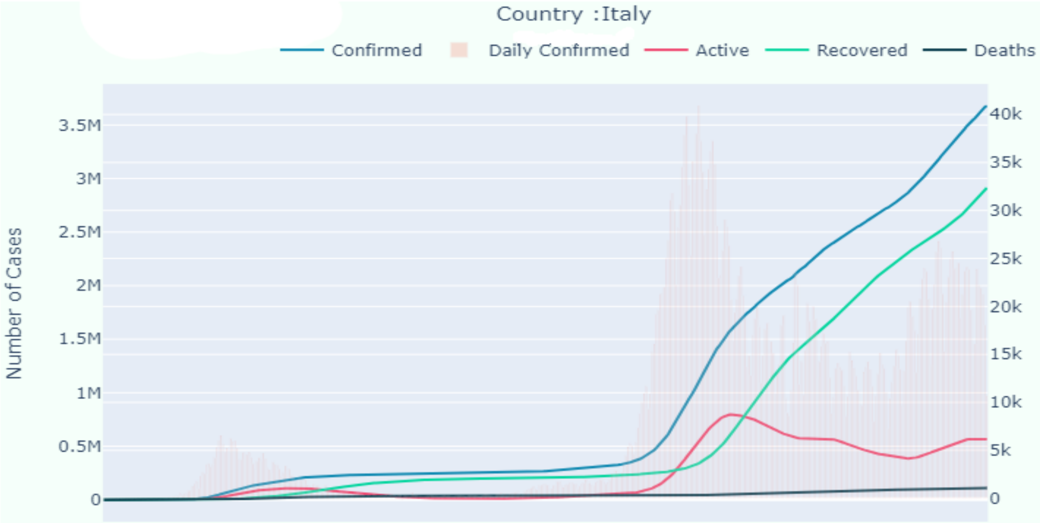


Figure 1. *Top 10 countries with the highest number of Covid-19 confirmed patients [19].*

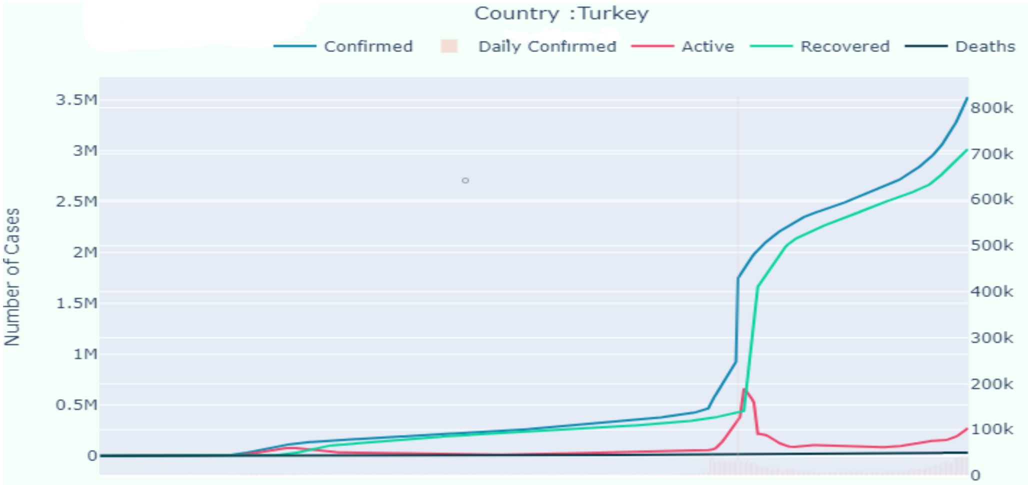
In Figure 1, 10 countries with the highest number of confirmed patients between January 22, 2020 and April 04, 2021 are given. Five countries with no missing data have been selected among the top 10 countries. These are India, Italy, Turkey, United Kingdom (UK) and United States (US). On the other hand, number of active cases, recovered cases, and death cases of five countries used in the study are given in Figure 2. Considering Figure 2, it can be seen that the number of the confirmed patients for five countries is increasing day by day. In India, Italy and Turkey, the number of recovered patients is increasing, while the number of active cases is increasing in the UK and US.



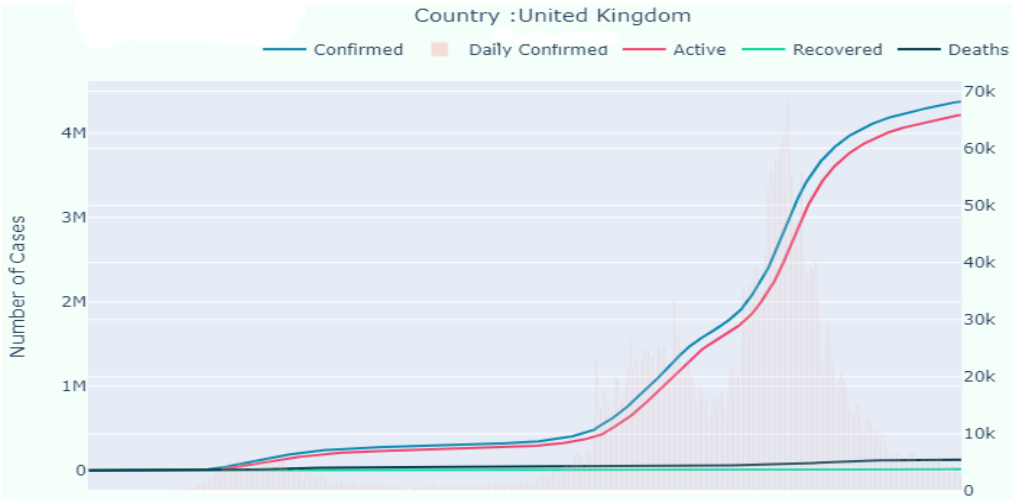
(a) *India*



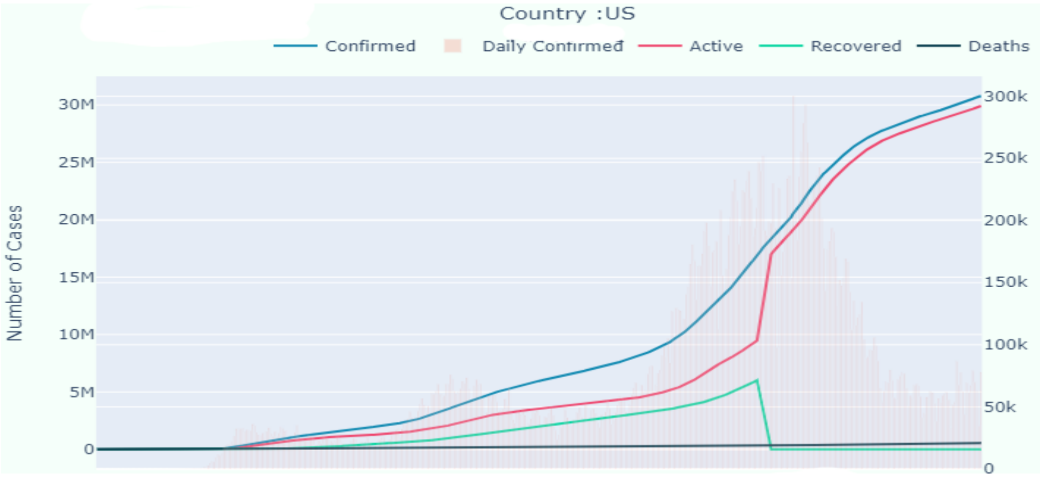
(b) *Italy*



(b) *Turkey*



(d) *United Kingdom*



***(e)** USA*

***Figure 2.** : Number of active, recovered, and death cases of countries used to predict the positive rate in the study [19].*

In this study, extreme learning machine algorithms were used to estimate the rate of positive cases. Accordingly, the prediction process with the extreme learning machine methods are performed by selecting the effective features with the F-test feature ranking method on the data. Different extreme learning algorithms (Extreme Learning Machine (ELM), Kernel ELM (three types as Linear, Radial-based and Polynomial ELM), Online Sequential ELM (OSELM), Constrained Sum ELM (CSELM)) for selected countries are presented comparatively.

The design of this article is given as follows: Section 2 describes the separately used ELM models. In Section 3, the characteristics of used data set, the F-test feature selection method and the parameters used for each ELM algorithm are given. The results of each method for each country are also interpreted and visualized with the help of graphics and tables in this section. Finally, in Section 4, the obtained results have been discussed and the studies planned have been mentioned.

II. METHODS AND METHODOLOGY

A. EXTREME LEARNING MACHINE

ELM is a single layer feed-forward network model with random input parameters. In ELM, weights between input and hidden layer are randomly generated [20]. The weights between the hidden and output layer are determined analytically. ELM produces successful results in regression and classification problems due to its analytical structure [21]. The input and output training sample for ELM are expressed as$(x_i, y_i) \in \mathbb{R}^n \times \mathbb{R}^m$,$(i = 1,2, \dots, N)$. The mathematical model of SLFN is as follows, where $\widehat{N}$ the number of hidden neurons and g is is the activation function.

$$\sum_{i=1}^{\widehat{N}} \alpha_i g(w_i . x_j + b_i) = o_j \tag{1}$$

where o_i is the i^{th} output value, N is number of train data, $w_i \in \mathbb{R}^n$ is a weight vector, bias is represented by b_i, and $\alpha_i = [\alpha_{i1}, \alpha_{i2}, \dots, \alpha_{im}]^T$ is the parameter vector between the hidden node and the output nodes. $w_i = [w_{i1}, w_{i2}, \dots, w_{in}]^T$ are randomly generated learning parameters between input and hidden layer. The mathematical model of SLFN approaching zero error is expressed by $\sum_{j=1}^{\widehat{N}} \parallel o_i - t_j \parallel = 0$.

$$\sum_{j=1}^{\widehat{N}} \alpha_i g(w_i . x_j + b_i) = t_j, j = 1,2, \dots, N \tag{2}$$

Equation 2 can be expressed in detail as

$$H\alpha = T \quad (3)$$

The left side of the equation can be expressed as $f(x) = h(x)\alpha$.
where,

$$H = \begin{bmatrix} g(w_1.x_1 + b_1) & g(w_{\hat{N}}.x_1 + b_{\hat{N}}) \\ \vdots & \vdots \\ g(w_1.x_N + b_1) & g(w_{\hat{N}}.x_N + b_{\hat{N}}) \end{bmatrix}_{N\times\hat{N}}, \quad (4)$$

$$\alpha = \begin{bmatrix} \alpha_1^T \\ \vdots \\ \alpha_{\hat{N}}^T \end{bmatrix}_{\hat{N}\times m} \quad and\ T = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix}_{N\times m}.$$

H represents the output matrix of the hidden layer. In ELM, the weights in the output layer are calculated analytically, where the w_j and b_j parameters are randomly assigned. It is required to solve the linear equation system given in Equation 3 to find the w_j and b_j parameters. Thus, the α vector forms the solution set of linear equations.

$$\hat{\alpha} = H^{\dagger}T \quad (5)$$

In Equation 5, $H^{\dagger}$ is the Moore-Penrose inverse of H. In fact, the output parameters updated in the ELM structure refer to the $\hat{\alpha}$ vector. Therefore, only the output weights are updated in the single hidden layer network. On the other hand, a regularized extreme learning machine (RELM) structure is proposed to provide stability and robustness in the ELM structure [22]. As a result of ELM, the standard deviation can be high due to randomness. Therefore, RELM takes into account heteroscedasticity and produces more stable results than ELM at outliers. RELM equation is expressed as Equation 6.

$$\hat{\alpha} = H^T\left(\frac{I}{C} + H^T H\right)^{\dagger} T \quad (6)$$

In this study, RELM is used because of its stable structure. C values are used systematically within the set of C^* where $C^* = \{2^{-10}, 2^{-9}, \ldots, 2^9, 2^{10}\}$. ELM only has a regularization parameter.

In ELM, weights and bias are updated analytically. However, the regularize parameter has been tested by selecting all values in the C* set. The ELM algorithm returns predicted and RMSE values. Although the weights and bias change during training, the regularized parameter does not change.

B. KERNEL EXTREME LEARNING MACHINE

Kernel extreme learning machines (KELM) algorithms have extremely fast learning and generalizing performances [23]. In this section, kernel based ELM algorithms are examined. The compact form of an ELM can be expressed by Equation 7.

$$f(x) = h(x)\alpha \quad (7)$$

Considering in Equation 7, its general form is given in Equation 8. The kernel matrix of extreme learning is defined as follows [21].

$$f(x) = h(x)H^T\left(\frac{I}{C}+H^TH\right)^{\dagger}T == \begin{bmatrix} K(x,x_1) \\ . \\ . \\ . \\ K(x,x_N) \end{bmatrix}^T \left(\frac{I}{C}+\omega_{ELM}\right)^{\dagger}T \tag{8}$$

where,

$$\omega_{ELM} = HH^T : \omega_{ELM_{i,j}} = h(x_i)h(x_j) = K(x_i,x_j) \tag{9}$$

If h(x) is unknown, Mercer conditions based on ELM can be defined. Equation 9 expresses these conditions. K represents the kernel. In this study, three kernel types are used. These are radial basis function (RBF) kernel, polynomial kernel and linear kernel. These kernel types are frequently used in the literature [24-27]. The equations of the three kernel types used are given below.

- Linear Kernel: $K(u,v) = u^Tv$
- RBF Kernel (Laplacian Kernel): $K(u,v) = exp\left(-\frac{\|u-v\|}{\sigma}\right)$
- Polynomial Kernel: $K(u,v) = (u^Tv + c)^d$

Linear kernel is the simplest kernel function, and it can be expressed as inner product. Laplacian kernel is a radial based kernel function. Choosing the sigma parameter is very important. If the sigma value is large, the kernel function starts to behave almost linearly. If the sigma value is taken too small, the training data will be highly sensitive to noise. The polynomial kernel is a non-constant kernel. Polynomial kernels are well suited for problems where all training data is normalized. c and d represent constant and polynomial degree respectively. The linear kernel has no individual parameters. The RBF kernel has a sigma parameter. There are c and d parameters in the polynomial kernel. The regularization parameter is common to all kernels.

C. CONSTRAINED SUM EXTREME LEARNING MACHINE

Since the weights between input and hidden layer are randomly assigned in ELM, it does not exhibit a stable structure. In constrained sum extreme learning machine (CSELM), the weights between input and hidden layer are created based on randomly selected sum vectors from the within-class sample [28]. CSELM initially chooses a random sample such as within-class $x_{c'}$ and $x_{c''}$. The sum of the two vectors $(x_{c'} + x_{c''})$ and the normalized vector sum are computed. The normalized sum vector is assigned to the weights between the input layer and the hidden layer. The normalized sum vector is calculated as,

$$w = \frac{x_{c'}+x_{c''}}{\|x_{c'}+x_{c''}\|_{L_2}^2} \tag{10}$$

where, L_2 norm is known as the Euclidean norm. With CSELM, random vectors are limited to a set of intraclass sum vectors. The structural design of CSELM is described below.

Let the bias and weights between the input layer and the hidden layer be represented as $b_{1\times\widehat{N}}$ and $w_{n\times\widehat{N}}$, respectively. The weight matrix between hidden and output layers is $\alpha_{\widehat{N}\times m}$

- Number of constrained sum vectors is chosen less than the number of neurons in the hidden layer ($< \widehat{N}$).
- Randomly selected vectors $x_{c'}$ and $x_{c''}$ from the same class. The sum vector $x_{c'} + x_{c''}$ is then constructed.
- Normalize the sum vector is calculated for input weights = $\left(\frac{x_{c'}+x_{c''}}{\|x_{c'}+x_{c''}\|_{L_2}^2}\right)$, while uniform distribution random numbers [0,1] are generated for bias (b).

• w and b are used for $b_{1\times\hat{N}}$ and $w_{n\times\hat{N}}$ vectors, respectively.
• The output matrix of the hidden layer is computed.

$$H = \begin{bmatrix} g(w_1.x_1 + b_1) & g(w_{\hat{N}}.x_1 + b_{\hat{N}}) \\ \vdots & \vdots \\ g(w_1.x_N + b_1) & g(w_{\hat{N}}.x_N + b_{\hat{N}}) \end{bmatrix}_{N\times\hat{N}}$$

• Output weights ($\alpha_{\hat{N}\times m}$) are calculated with least square estimation. $\hat{\alpha} = H^{\dagger}T$, where $H^{\dagger}$ represents pseudo inverse.
CSELM only has a regularization parameter.

D. ONLINE SEQUENTIAL EXTREME LEARNING MACHINE

Online sequential extreme learning machine (OSELM) is a successful example of a batch-based learning system. The structure of OSELM was developed for single layer feed forward neural network based on ELM [29].

The input and output training sample for OSELM are expressed as $(x_i, t_i) \in \mathbb{R}^n \times \mathbb{R}^m$,$(i = 1,2, \dots, N)$, The mathematical model of SLFN is expressed as,

$$f_{\hat{N}}(x_j) = \sum_{i=1}^{\hat{N}} \alpha_i g(a_i, b_i, x_j) = t_j \tag{11}$$

where $\hat{N}$ and $j = 1,2, \dots N$ is the number of hidden neurons and g is the activation function. a_i and b_i are the weights in the hidden layer. α_i is the output weight. $g(a_i, b_i, x_j)$ is the output of the ith hidden node for the jth input. The ELM structure can be inspired by the mini batch method to create an OSELM structure.

Suppose that data $\aleph = \{(x_i, t_i)|x_i \in \mathbb{R}^n, t_i \in \mathbb{R}^m\}$ is expressed sequentially (one-by-one or chunk-by-chunk). OSELM has two main phases. These are initialization and sequential phase. if $rank(H_0) = \hat{N}$, OSELM and ELM perform the same learning performance. H_0 represents the output of the hidden layer for the initialization phase. In the initial phase, the training data size must be greater than or equal to the number of hidden nodes ($N_0 \geq \hat{N}$).

• Step 1: Initialization Phase Learning is practised using a small chunk of the training set.
Let the small chunk $\aleph_0 = \{(x_i, t_i)\}_{i=1}^{N_0}$ taken in the $\aleph = \{(x_i, t_i)|x_i \in \mathbb{R}^n, t_i \in \mathbb{R}^m\}$
1. a_i and b_i are randomly assigned according to the uniform distribution.
2. Output matrix of the hidden layer is calculated H_0.

$$H_0 = \begin{bmatrix} g(w_1.x_1 + b_1) & g(w_{\hat{N}}.x_1 + b_{\hat{N}}) \\ \vdots & \vdots \\ g(w_1.x_N + b_1) & g(w_{\hat{N}}.x_N + b_{\hat{N}}) \end{bmatrix}_{N\times\hat{N}}$$

3. The initial output parameter is calculated.

$\alpha^{(0)} = P_0 H_0^T T_0$ and $P_0 = (H_0^T H_0)^{-1}, T_0 = [t_1, \dots, t_{N_0}]^T$ 4. Set k=0 (k represents the number of chunks in the data.)

• Step 2: Sequential learning Phase Present the (k+1)th chunk of new observations
$\aleph_{k+1} = \{(x_i, t_i)\}_{i=(\sum_{j=0}^{k} N_j)+1}^{\sum_{j=0}^{k+1} N_j}$ and N_{k+1} represents the number of observations in the (k+1)th chunk.

1. For the (k+1)th chunk of $\aleph_{k+1}$ data, the partial hidden layer output matrix is calculated with equation 12

2. Output weights are calculated ($\alpha^{(k+1)}$).

$$P_{k+1} = P_k - P_k H_{k+1}^T (I + H_{k+1} P_k H_{k+1}^T)^{-1} H_{k+1} P_k$$
$$\alpha^{(k+1)} = \alpha^k + P_{k+1} H_{k+1}^T (T_{k+1} - H_{k+1} \alpha^{(k)}).$$

3. Set $T_{k+1} = \left[t_{(\sum_{j=0}^{k} N_j)+1}, \ldots, t_{\sum_{j=0}^{k+1} N_j}\right]^T$

4. Set k=k+1 Return to the beginning of the sequential learning phase.

$$H_{k+1} = \begin{bmatrix} g(w_1.x_{\sum_{j=0}^{k} N_j} + b_1) & g(w_{\hat{N}}.x_{\sum_{j=0}^{k} N_j} + b_{\hat{N}}) \\ \vdots & \vdots \\ g(w_1.x_{\sum_{j=0}^{k} N_j} + b_1) & g(w_{\hat{N}}.x_{\sum_{j=0}^{k} N_j} + b_{\hat{N}}) \\ & \end{bmatrix}_{N_{k+1} \times \hat{N}} \tag{12}$$

In OSELM, the chunk size does not have to be constant, it can take values in varying sizes. OSELM has two important parameters. Firstly, N0: Number of initial training data used in the initial phase of OSELM, which is not less than the number of hidden neurons. Secondly, block: Size of block of data learned by OSELM in each step.

In ELM, input parameters are randomly assigned, and output parameters are calculated analytically, and only the weights between the hidden layer and the output layer are learned. However, the randomness of the input values can significantly change the results. In addition, the activation function in the hidden layer also affects the operation of ELMs, and a nonlinear system is formed through the nonlinear activation functions in the hidden layer [30]. Therefore, Kernel ELM is both faster and more stable. It has a single hidden layer and the activation functions in the hidden layer are kernel-based. Kernel-based functions are faster than others. In CSELM, the randomly selected inputs of the ELM are designed with a constrained sum. Batch learning system is applied in OSELM algorithm. With this system learning is implemented chunk by chunk. Generally speaking, ELM-based methods are not iterative-based allows them to reach results much faster than traditional algorithms. A detailed study on the comparison of ELM with other methods in the literature was done by Wang et al [30].

III. EXPERIMENTAL STUDY and DATASET

The data set used in the study is a collection of Covid-19 data and is updated daily [31]. This dataset is particularly suitable for researchers working on predicting data such as confirmed cases, mortality rate and future cases. In the presented study, five countries (India, Italy, Turkey, UK and USA) are discussed considering the date range specified at he beginning of the study. Since the Covid-19 pandemic has emerged, various data on this field have been used, e.g. time series data and image data. In this study, extreme learning algorithms are applied to current Covid-19 data. Firstly, a pre-processing has been applied on the data. Then, parameters of each method and experiment methodology are given. The data have been selected based on the occupancy rate of the data for each country and the feature selection method. The features in the data are new_cases, new_deaths, reproduction_rate, icu_patients, hosp_patients, new_tests, stringency_index and positive_rate.

The meanings of these properties are as follows:

• *stringency_index:* frequency of intervention determined by the government, takes a value in the range of 0-100, includes situations such as school and workplace closures and travel bans.

• *new_cases:* newly confirmed cases of Covid-19.

• *new_tests:* new tests for Covid-19 (calculated for consecutive days only)

• *reproduction_rate:* real-time prediction information of the effective reproduction rate (R) of Covid-19.

• *new_deaths:* new deaths attributed to Covid-19.

• *icu_patients:* the number of Covid-19 patients in intensive care units (ICU) on a given day.

• *hosp_patients:* the number of Covid-19 patients hospitalized on a given day.

• *positive_rate:* the positive part of the Covid-19 tests given as an average of 7 days.

Table 1 gives a sample of data for 7 features for UK. The data was normalized by applying Equation 13 on the original data. This process is applied for all country data.

Table 1. *Normalized sample data for UK.*

stringency_index	new_cases	new_tests	reproduction_rate	new_deaths	icu_patients	hosp_patients
0.00137	0.01424	0.21359	0.02315	0.06775	0.14980	0.00600
0.00298	0.01314	0.20388	0.02041	0.08620	0.14980	0.00600
0.00357	0.01424	0.22330	0.02066	0.10510	0.14980	0.00600
0.00567	0.00493	0.27184	0.01917	0.06487	0.14980	0.00600
0.00415	0.00602	0.21359	0.01767	0.03461	0.14980	0.00600
0.00208	0.00548	0.13592	0.01718	0.00415	0.14980	0.00600
0.00000	0.01369	0.08252	0.01917	0.06045	0.14980	0.00500
0.00174	0.00931	0.15049	0.01245	0.06691	0.14980	0.00500
0.00493	0.00493	0.26214	0.01270	0.07237	0.14980	0.00500
0.00478	0.01752	0.27184	0.01095	0.08103	0.14980	0.00500

F-test feature ranking used for the selection of the relevant feature in the study is a filter method based on the F statistical test. Filter methods takes into account the relationship between attributes and target to calculate the attributes significance. In this study, positive rate was chosen as the target. Therefore, F-test feature ranking is calculated by considering whether there is a significant difference between the positive rate and one of each other features.

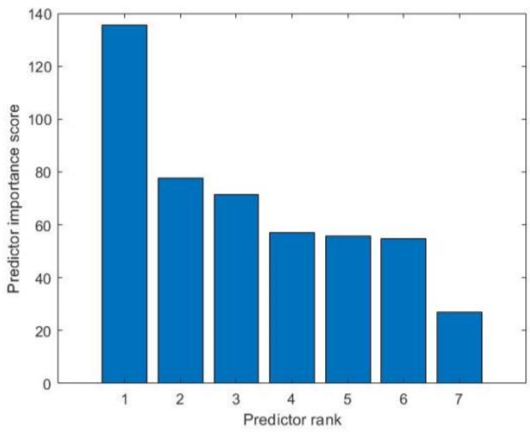


Figure 3. *Feature selection according to F-test feature ranking.*

Table 2. *Parameter sets for extreme learning methods.*

Extreme Learning Methods	Parameter Types	Parameter Set
ELM and CSELM	C N	$[-2^{10}, 2^{10}]$ 10,20,50,75,100
RBF_kernel	C Gamma	$[-2^{10}, 2^{10}]$ $[-2^{10}, 2^{4}]$
POLY_kernel	C c1 d	$[-2^{10}, 2^{10}]$ 0:0.1:1 d:1:10
LIN_kernel	C	$[-2^{10}, 2^{10}]$
OSELM	N_0≥N Block size	10,20,50,75,100 one-by-one

The result of the feature selection method is given in Figure 3. The indexes of the features are obtained as 7, 1, 6, 3, 2, 4, 5 respectively. Therefore, in Figure 3, 1 represents feature 7, while 7 represents feature 5. While the feature with the highest predictor rank is stringency_index, the feature with the lowest predictor rank is obtained as hosp_patients. Since the hosp_patients feature is significantly lower among the others, it has been removed from the feature list. In Turkey and India, icu_patients data is incomplete, it is removed from feature set*. According to the above mentioned, the number of train and test data and features of the data sets is summarized in Table 3.

Table 3. *Properties of the data sets by the countries*

Countries	Number of Train and Test Data	Number of Features
India	(150,66)	6*
Italy	(150,66)	7
Turkey	(150,66)	6*
UK	(150,66)	7
USA	(150,66)	7

Since the data are in different ranges, max-min normalization is applied. The max-min normalization is calculated as

$$Data_{norm}(i,j) = \frac{X(i,j) - X_{min}(i,j)}{X_{max}(i,j) - X_{min}(i,j)} \tag{13}$$

where $i = 1,2,\cdots$,number of days, $j = 1,2,\cdots$,number of features. X_{min} represents the min value and X_{max} represents the max value of the same feature. X represents the feature of the related day.

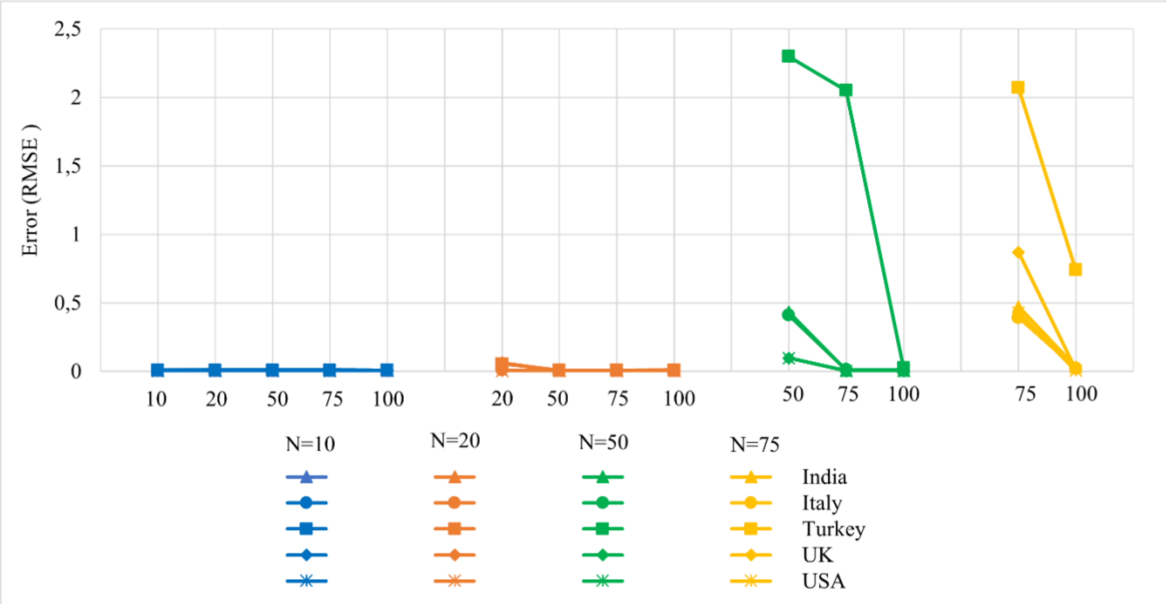


Figure 4. *RMSE results according to the N and N_0 parameters of OSELM.*

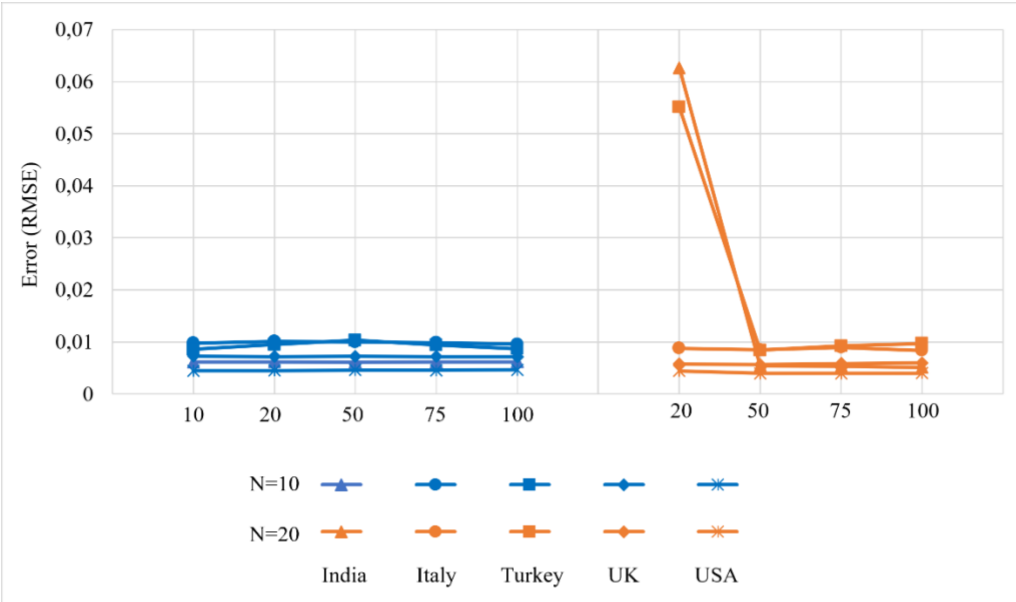


Figure 5. *Zoom of RMSE results with respect to N_0 values for N = 10 and N = 20.*

The specific parameters of the extreme learning algorithms play an important role in predicting the positive patient rate of countries. The set of parameters used for each extreme learning method is given in Table 2.

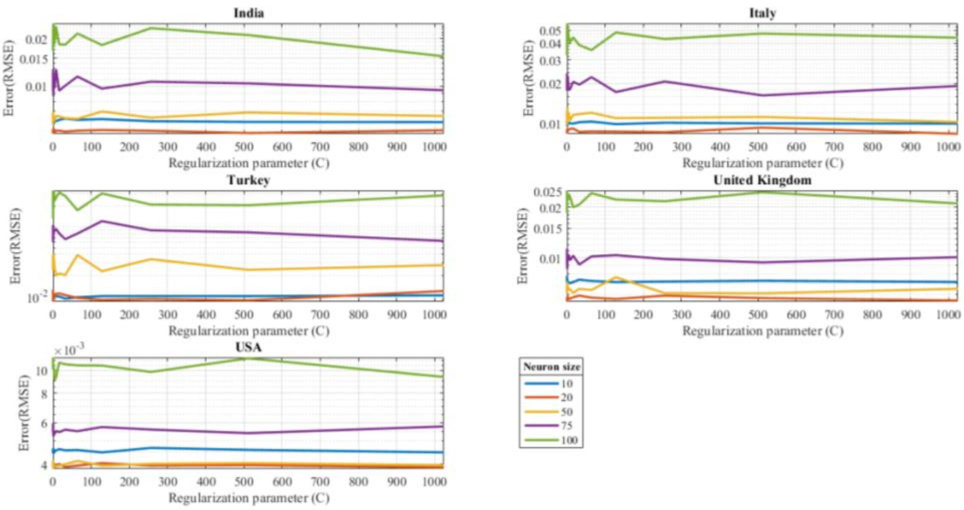


Figure 6. *ELM RMSE results according to the regularization parameter based on neuron size.*

For Turkey and India, input-output vector is represented by [new_cases new_deaths reproduction_rate new_tests stringency_index; positive_rate]. For Italy, UK and US, input-output vector is represented by [new_cases new_deaths reproduction_rate icu_patients new_tests stringency_index; positive_rate].
Six different extreme learning algorithms have been compared to predict the Covid-19 positive rate of five countries. After 50 independent runs, the RMSE results have been averaged. RMSE can be calculated as,

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (Predicted_i - Actual_i)^2}{N}} \tag{14}$$

where, $Predicted_i$: The predicted value for the ith observation, $Actual_i$: The actual value for the ith observation, N: Total number of observations. The main goal is to find the ideal parameter set of ELM methods for each country. As a result of the experiments, the sigmoid activation function has been used in ELM, CSELM and OSELM methods. Firstly, we have discussed the ELM structure. Since it is the basic structure of extreme learning architecture. RMSE results for five countries are given in Figure 6. Considering that, the lowest RMSE value of ELM neuron size is 20. It can be seen that the optimal neuron size for all countries is 20. On the other hand, the regularization parameter differs from country to country. The ideal regularization parameter for India can be chosen between 400 and 600. Also, ideal regularization parameter for Turkey can be chosen between 200 and 400. For other countries (Italy, UK, USA), the ideal regularization parameter 1024 can be selected.

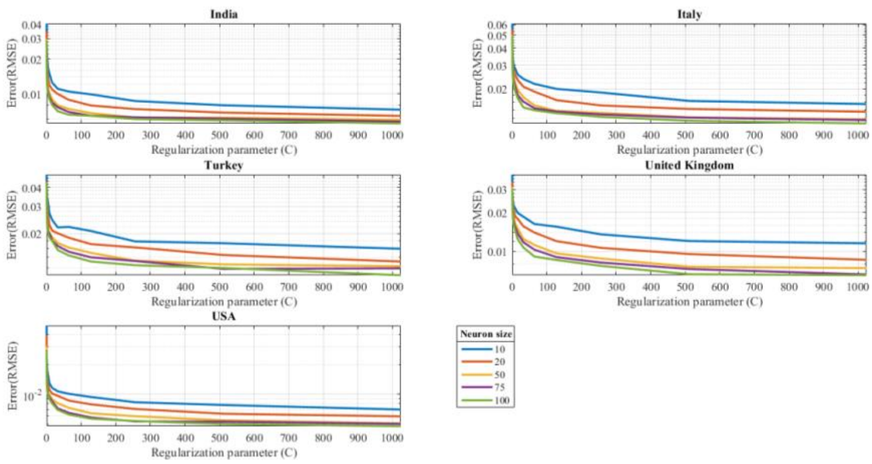


Figure 7. *CSELM RMSE results according to the regularization parameter based on neuron size.*

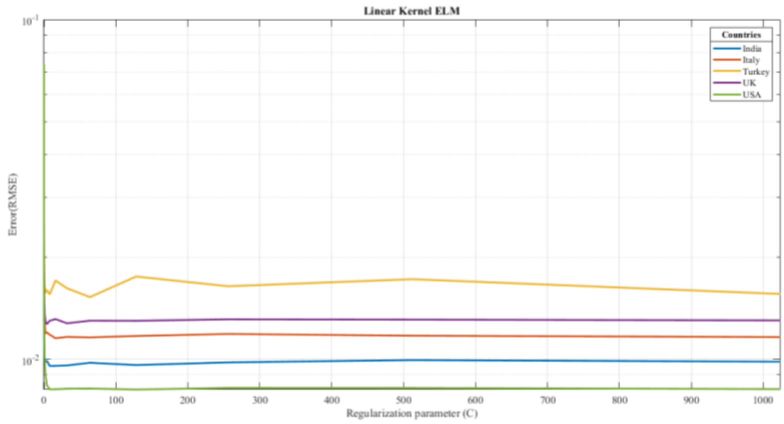
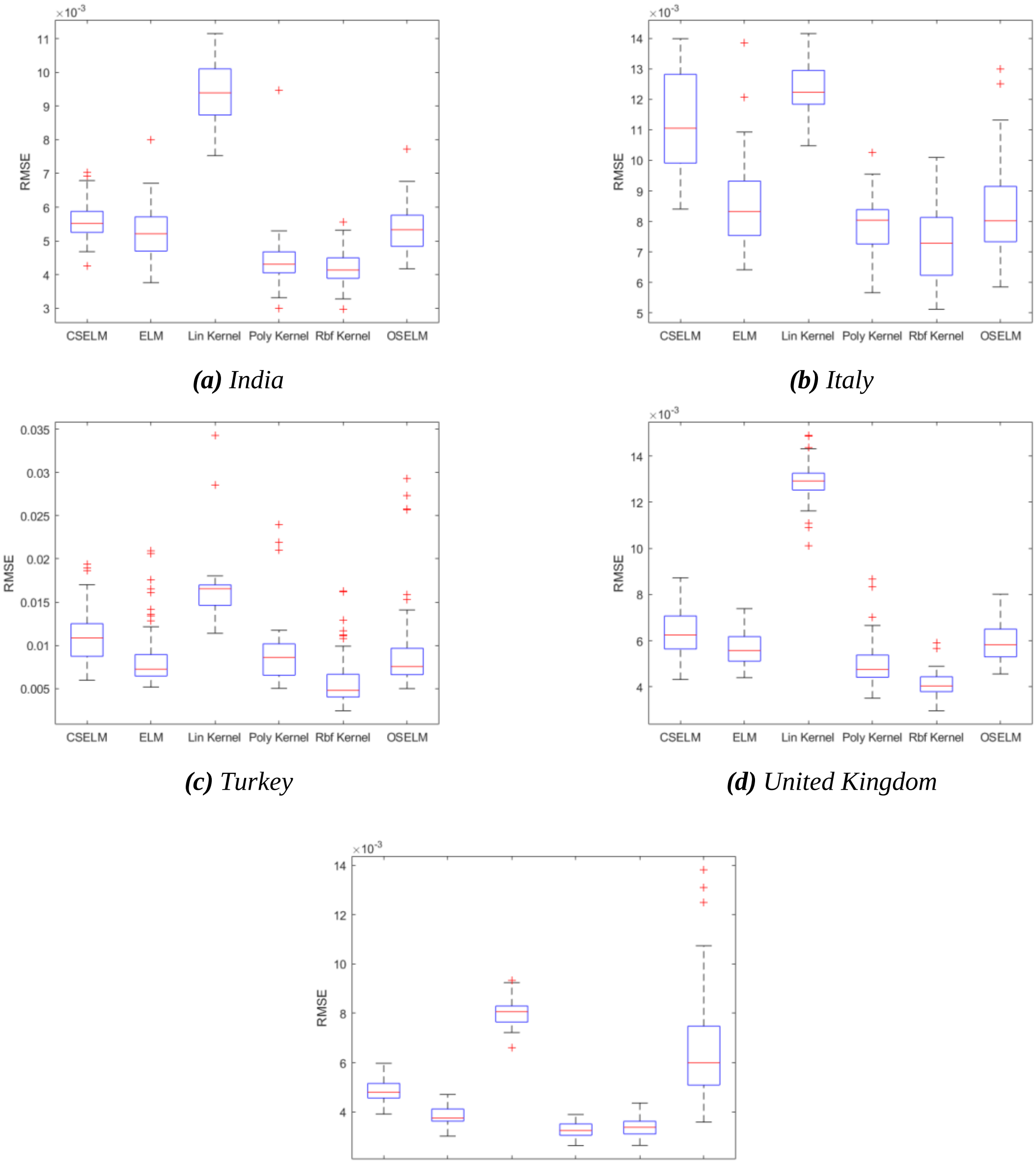


Figure 8. *Linear Kernel RMSE results according to the regularization parameter.*

In Figure 4, RMSE results are given according to the N0 and N values of OSELM. Experiments were applied as $N_0 \geq N$. Also, Figure 5 shows zoomed analysis of N0 values for N = 10 and N = 20. N and N0 parameters are chosen [20, 100] and [20, 50] and [50, 75] respectively, for India, Italy and Turkey, UK and US ([N,N_0]). India and Italy have the same set of parameters. Similarly, UK and US have the same set of parameters.

In ELM, RMSE results vary because input parameters are randomly assigned. However, CSELM handles input parameters with a specific method. Therefore, CSELM produces more stable and consistent results than ELM. In Figure 7, CSELM RMSE results are given according to the regularization parameter based on neuron size. Considering that, it can be seen that the RMSE value decreases as the regularization parameter increases. It is also clear that the ideal neuron size for all countries is 100. In Figure 8, RMSE results of linear kernel based ELM method according to regularization parameter are given. In linear kernel ELM, the effect of regularization parameter is less than other methods. It can be seen that other countries are more stable. If Figure 8 is examined, the regularization parameter can be chosen as: India: 128, Italy: 2, Turkey: 64, UK:4 , USA: 1024.



(a) India *(b) Italy*

(c) Turkey *(d) United Kingdom*

(e) USA

Figure 9. *Comparison of ELM methods for each country.*

RBF and polynomial kernel ELMs have a large number of parameter permutations. This situation appears more clearly in Table 2. Therefore, the 10 best results for parameter settings of RBF and polynomial kernels are tabulated. Table 4 shows 10 cases of polynomial and RBF kernels with the most ideal parameters for each country.

__Table 4.__ Parameter settings based on top 10 results for RBF and Poly.

	RBF_INDIA			POLY_INDIA		
C	**Gamma**	**RMSE**	**C**	**C1**	**d**	**RMSE**
128	**1**	**4.17E-03**	**32**	**0.6**	**5**	**4.28E-03**
512	1	4.19E-03	8	0.9	5	4.28E-03
64	1	4.20E-03	8	0.8	5	4.29E-03
512	2	4.21E-03	1	0.9	6	4.30E-03
256	1	4.22E-03	4	0.9	6	4.34E-03
1024	1	4.24E-03	2	1	6	4.34E-03
128	0.5	4.24E-03	4	1	5	4.34E-03
1024	2	4.25E-03	8	1	5	4.34E-03
32	0.5	4.29E-03	64	0.9	4	4.36E-03
64	0.5	4.32E-03	16	0.7	5	4.39E-03
	RBF_ITALY			**POLY_ITALY**		
C	**Gamma**	**RMSE**	**C**	**C1**	**d**	**RMSE**
256	**0.5**	**6.87E-03**	**128**	**0.1**	**3**	**7.29E-03**
256	1	7.13E-03	32	0.2	2	7.40E-03
128	1	7.19E-03	4	0.9	3	7.44E-03
128	0.5	7.28E-03	16	0.2	4	7.45E-03
512	2	7.31E-03	8	0.4	3	7.51E-03
64	0.5	7.31E-03	32	0.1	3	7.52E-03
64	1	7.31E-03	8	0.5	3	7.54E-03
32	0.5	7.34E-03	16	0.2	3	7.58E-03
16	1	7.36E-03	32	0.2	3	7.58E-03
64	2	7.37E-03	8	1	4	7.59E-03
	RBF_UK			**POLY_UK**		
512	**1**	**4.07E-03**	**128**	**1**	**4**	**4.70E-03**
1024	1	4.25E-03	32	1	5	4.82E-03
256	1	4.31E-03	128	0.8	4	4.83E-03
1024	0.5	4.32E-03	128	0.9	4	4.84E-03
128	0.5	4.34E-03	256	1	4	4.86E-03
64	0.5	4.41E-03	128	1	2	4.90E-03
512	0.5	4.41E-03	64	0.8	2	4.90E-03
256	0.5	4.44E-03	128	0.7	2	4.91E-03
128	1	4.57E-03	32	0.8	2	4.92E-03
64	1	4.63E-03	64	0.7	2	4.93E-03
	RBF_US			**POLY_US**		
1024	**2**	**3.35E-03**	**64**	**0.8**	**3**	**3.26E-03**
1024	4	3.36E-03	64	0.7	3	3.28E-03
512	2	3.40E-03	128	0.6	3	3.28E-03
512	4	3.45E-03	64	1	3	3.28E-03
256	2	3.48E-03	32	0.9	3	3.29E-03
512	1	3.50E-03	32	1	3	3.30E-03
1024	8	3.50E-03	32	0.7	3	3.30E-03
128	1	3.53E-03	128	0.5	3	3.31E-03
128	2	3.54E-03	16	0.9	3	3.31E-03
256	1	3.61E-03	64	0.4	3	3.31E-03
	RBF_TU			**POLY_TU**		
256	**0.25**	**4.96E-03**	**8**	**1**	**3**	**7.60E-03**
128	0.25	5.11E-03	16	0.9	2	7.79E-03
512	0.5	5.47E-03	4	0.9	4	7.86E-03
64	0.25	5.61E-03	64	0.5	2	7.92E-03
64	0.125	5.81E-03	4	1	3	7.94E-03
512	0.25	5.87E-03	32	0.9	2	8.02E-03
128	0.5	5.92E-03	2	0.8	4	8.05E-03
16	0.5	6.05E-03	64	0.7	2	8.05E-03
32	0.25	6.13E-03	16	0.5	3	8.05E-03
16	0.25	6.14E-03	64	0.6	2	8.12E-03

Considering all the parameter analysis, there are different parameter values for each method and each country. In the aforementioned experiments, the most ideal parameter sets were determined according to each ELM method. Based on ideal parameters, the average RMSE values were obtained after 50 independent runs of six different ELM methods. The results obtained were figured with box-plot.

In Figure 9, RBF gives the lowest RMSE result in all countries except for US. If Figure 9c is examined carefully, it can be seen that the country with the most outlier data is Turkey. The reason for this situation is due to the sudden increase in the number of Covid-19 cases. With the sudden increase in the rate of Covid-19 in a certain period in Turkey, the standard deviation of the positive case increases. After the RBF kernel, the most ideal structure is the polynomial kernel-based ELM method. It can be said that the worst performing ELM method is a linear kernel.

IV. CONCLUSION AND FUTURE WORKS

In this study, six different ELM methods based on five features have been used to estimate the Covid-19 positive rate. The optimal parameters of each ELM method were calculated. Then, parameter analysis was performed for each country in the article. Considering Figure 9, it can be said that the data for India is more consistent than other countries. On the contrary, it is clear that the standard deviation of the data for Turkey is higher than that of other countries. The new cases, new deaths, reproduction rate, intensive care unit patients, new tests and stringency index features are used to predict the positive rate. The most frequently used ELM methods (ELM, Kernel ELM (Linear, Radial basis and Polynomial ELM), OSELM, CSELM) are considered in the study. These ELM methods have been compared to estimate the Covid-19 positive rate. Within the scope of the study, up-to-date Covid-19 data have been used. Also, a deep parameter analysis has been performed for all ELM methods. The ELM method that predicts the Covid-19 positive rate with the least error is RBF kernel. Also, as a result of the experiments, it can be seen that the ELM method with the highest error is polynomial kernel-based ELM.

For future works, it is planned to apply the RBF Kernel ELM algorithm to predict positive rates for the next 1 week or 1 month. Also, by using RBF, studies can be conducted on modelling different features in the database (such as mortality rate, confirmed cases, outcome prediction). Moreover, comparative studies can be presented by applying RBF-based ELM models for different time series Covid-19 data. As applied in the study [13], improving the ELM model parameters with the use of optimization algorithms can also be considered for future studies

V. REFERENCES

[1] WHO. (2020, 2021-04-26). *World health organization (2020) covid-19 situation reports*.

Available: https: //www.who.int/emergencies/diseases/novel-coronavirus-2019/ situation-reports.

[2] Worldometer, "Coronavirus cases:," 2021-04-26.

[3] Q. Li, W. Feng, and Y.-H. Quan, "Trend and forecasting of the COVID-19 outbreak in China," *Journal of Infection,* vol. 80, no. 4, pp. 469-496, 2020.

[4] D. Fanelli and F. Piazza, "Analysis and forecast of COVID-19 spreading in China, Italy and France," *Chaos, Solitons & Fractals,* vol. 134, p. 109761, 2020.

[5] W. Wei *et al.*, "Application of a combined model with autoregressive integrated moving average (ARIMA) and generalized regression neural network (GRNN) in forecasting hepatitis incidence in Heng County, China," *PloS one,* vol. 11, no. 6, p. e0156768, 2016.

[6] Z. Ceylan, "Estimation of COVID-19 prevalence in Italy, Spain, and France," *Science of The Total Environment,* vol. 729, p. 138817, 2020.

[7] A. F. Lukman, R. I. Rauf, O. Abiodun, O. Oludoun, K. Ayinde, and R. O. Ogundokun, "COVID-19 prevalence estimation: Four most affected African countries," *Infectious Disease Modelling,* vol. 5, pp. 827-838, 2020.

[8] A. M. Almeshal, A. I. Almazrouee, M. R. Alenizi, and S. N. Alhajeri, "Forecasting the spread of COVID-19 in Kuwait using compartmental and logistic regression models," *Applied Sciences,* vol. 10, no. 10, p. 3402, 2020.

[9] R. O. Ogundokun, A. F. Lukman, G. B. Kibria, J. B. Awotunde, and B. B. Aladeitan, "Predictive modelling of COVID-19 confirmed cases in Nigeria," *Infectious Disease Modelling,* vol. 5, pp. 543-548, 2020.

[10] M. Djeddou, I. A. Hameed, A. Hellal, and A. Nejatian, "Predictive modeling of COVID-19 New Confirmed Cases in Algeria using Artificial Neural Network," *medRxiv,* 2021.

[11] M. A. Achterberg, B. Prasse, L. Ma, S. Trajanovski, M. Kitsak, and P. Van Mieghem, "Comparing the accuracy of several network-based COVID-19 prediction algorithms," *International journal of forecasting,* 2020.

[12] W. He, G. Y. Yi, and Y. Zhu, "Estimation of the basic reproduction number, average incubation time, asymptomatic infection rate, and case fatality rate for COVID-19: Meta-analysis and sensitivity analysis," *Journal of medical virology,* vol. 92, no. 11, pp. 2543-2550, 2020.

[13] A. M. Hasan, A. G. Mahmoud, and Z. M. Hasan, "Optimized Extreme Learning Machine for Forecasting Confirmed Cases of COVID-19," *International Journal of Intelligent Engineering and Systems,* pp. 484-494, 2021.

[14] G. Pinter, I. Felde, A. Mosavi, P. Ghamisi, and R. Gloaguen, "COVID-19 pandemic prediction for Hungary; a hybrid machine learning approach," *Mathematics,* vol. 8, no. 6, p. 890, 2020.

[15] M. Yadav, M. Perumal, and M. Srinivas, "Analysis on novel coronavirus (COVID-19) using machine learning methods," *Chaos, Solitons & Fractals,* vol. 139, p. 110050, 2020.

[16] S. Rath, A. Tripathy, and A. R. Tripathy, "Prediction of new active cases of coronavirus disease (COVID-19) pandemic using multiple linear regression model," *Diabetes & Metabolic Syndrome: Clinical Research & Reviews,* vol. 14, no. 5, pp. 1467-1474, 2020.

[17] S. Ghosal, S. Sengupta, M. Majumder, and B. Sinha, "Linear Regression Analysis to predict the number of deaths in India due to SARS-CoV-2 at 6 weeks from day 0 (100 cases-March 14th 2020)," *Diabetes & Metabolic Syndrome: Clinical Research & Reviews,* vol. 14, no. 4, pp. 311-315, 2020.

[18] R. Sujath, J. M. Chatterjee, and A. E. Hassanien, "A machine learning forecasting model for COVID-19 pandemic in India," *Stochastic Environmental Research and Risk Assessment,* vol. 34, pp. 959-972, 2020.

[19] therealcyberlord, "Coronavirus (covid-19) visualization & prediction."

[20] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing,* vol. 70, no. 1-3, pp. 489-501, 2006.

[21] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics),* vol. 42, no. 2, pp. 513-529, 2011.

[22] W. Deng, Q. Zheng, and L. Chen, "Regularized extreme learning machine," in *2009 IEEE symposium on computational intelligence and data mining*, 2009, pp. 389-395: IEEE.

[23] G.-B. Huang and C.-K. Siew, "Extreme learning machine: RBF network case," in *ICARCV 2004 8th Control, Automation, Robotics and Vision Conference, 2004.*, 2004, vol. 2, pp. 1029-1036: IEEE.

[24] W.-Y. Deng, Q.-H. Zheng, and Z.-M. Wang, "Cross-person activity recognition using reduced kernel extreme learning machine," *Neural Networks,* vol. 53, pp. 1-7, 2014.

[25] A. Iosifidis and M. Gabbouj, "On the kernel extreme learning machine speedup," *Pattern Recognition Letters,* vol. 68, pp. 205-210, 2015.

[26] Z. Bai, G.-B. Huang, D. Wang, H. Wang, and M. B. Westover, "Sparse extreme learning machine for classification," *IEEE transactions on cybernetics,* vol. 44, no. 10, pp. 1858-1870, 2014.

[27] K. Parikh and T. Shah, "Kernel based extreme learning machine in identifying dermatological disorders," *International Journal of Innovative Science, Engineering & Technology,* vol. 3, no. 10, pp. 370-375, 2016.

[28] W. Zhu, J. Miao, and L. Qing, "Constrained extreme learning machines: A study on classification cases," *arXiv preprint arXiv:1501.06115,* 2015.

[29] N.-Y. Liang, G.-B. Huang, P. Saratchandran, and N. Sundararajan, "A fast and accurate online sequential learning algorithm for feedforward networks," *IEEE Transactions on neural networks,* vol. 17, no. 6, pp. 1411-1423, 2006.

[30] J. Wang, S. Lu, S.-H. Wang, and Y.-D. Zhang, "A review on extreme learning machine," *Multimedia Tools and Applications,* pp. 1-50, 2021.

[31] J. Hasell *et al.*, "A cross-country database of COVID-19 testing," *Scientific data,* vol. 7, no. 1, pp. 1-7, 2020.