

Research Article

Evolutionary Computation Techniques for Predicting Atmospheric Corrosion

Amine Marref, Saleh Basalamah, and Rami Al-Ghamdi

Department of Computer Science, Umm Al-Qura University, Makkah 21955, Saudi Arabia

Correspondence should be addressed to Amine Marref; ajmarref@uqu.edu.sa

Received 23 November 2012; Accepted 21 January 2013

Academic Editor: Flavio Deflorian

Copyright © 2013 Amine Marref et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Corrosion occurs in many engineering structures such as bridges, pipelines, and refineries and leads to the destruction of materials in a gradual manner and thus shortening their lifespan. It is therefore crucial to assess the structural integrity of engineering structures which are approaching or exceeding their designed lifespan in order to ensure their correct functioning, for example, carrying ability and safety. An understanding of corrosion and an ability to predict corrosion rate of a material in a particular environment plays a vital role in evaluating the residual life of the material. In this paper we investigate the use of genetic programming and genetic algorithms in the derivation of corrosion-rate expressions for steel and zinc. Genetic programming is used to automatically evolve corrosion-rate expressions while a genetic algorithm is used to evolve the parameters of an already engineered corrosion-rate expression. We show that both evolutionary techniques yield corrosion-rate expressions that have good accuracy.

1. Introduction

Corrosion is a natural phenomenon that can cause substantial economic and environmental losses which result from the damage incurred in metal constructions over the years. The cost of corrosion has been reported [1, 2] to be as large as 3.1% of the gross domestic product of countries such as the United States, United Kingdom, and Australia. Corrosion costs can be (i) direct when the metallic structure is greatly damaged in which case replacement or expensive maintenance are required or (ii) indirect when the worsened appearance of the construction reduces its value (even if the construction is not greatly damaged and can still be used just fine).

Corrosion refers to the disintegration of materials into their constituent atoms because of chemical or electrochemical reactions with the environment [3]. This disintegration causes a loss in the thickness of the construction which results in a decrease in resistance and strength and consequently a decrease in the service performance of the construction. Corrosion occurs in many engineering structures such as bridges, pipelines, refineries, and so forth and can result in the destruction of materials in a gradual manner and hence shortening their lifespan.

Corrosion can occur in many environments such as atmosphere, soil, sea, and so forth where environmental factors affect the material in complicated processes leading to its corrosion. Depending on the environment, corrosion can be atmospheric, underground, marine, gaseous, or microbial and bacterial. Atmospheric corrosion is the type of corrosion we are mainly interested in because (i) it has been reported that atmospheric corrosion is responsible for more corrosion-induced failures than any other corrosion type [4] and (ii) it is the most major corrosion type in SABIC [5] industrial sites—where the findings of this work are going to be applied.

Because of its huge impact on the economy and environment, an understanding of corrosion and ability to predict corrosion rate of a material in a particular environment plays a vital role in evaluating the residual life of the material and consequently reducing associated costs. In order to understand and predict corrosion, we must model the environmental factors that influence corrosion and derive relationships between them and the rate of the resulting corrosion.

In this paper, we propose the use of genetic programming [6] and genetic algorithms [7], to derive the corrosion-rate

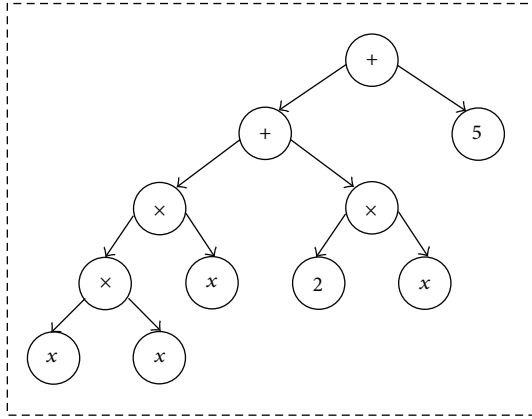


FIGURE 1: An example abstract-syntax tree corresponding to some function $f_i(x) = x^3 + 2x + 5$.

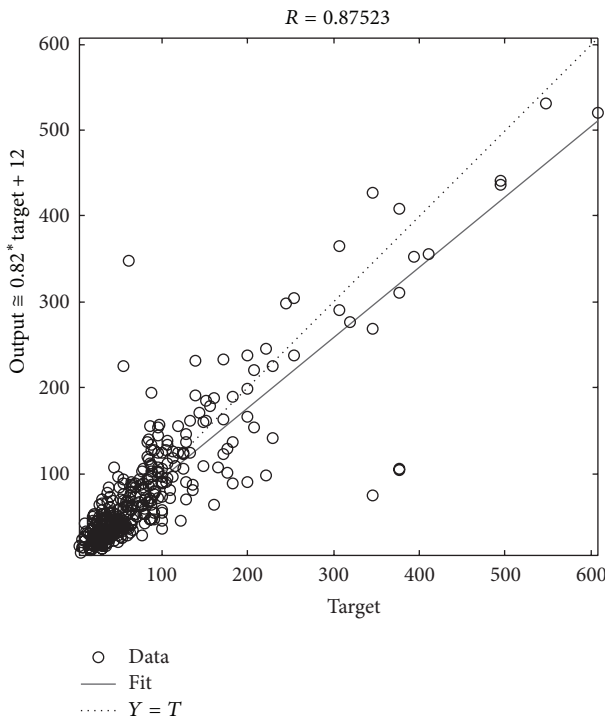


FIGURE 2: GP steel expression no. 1.

expressions in terms of the major influential environmental factors.

The rest of the paper is structured as follows. In Section 2, we will found the mathematical properties of the problem of deriving corrosion-rate expression. In Section 3, we will describe the methodology of our work where we use genetic programming and genetic algorithms. In Section 4, we conduct an empirical evaluation of our work. In Section 5 we discuss our findings. In Section 6 we review related work in the automatic derivation of corrosion-rate expressions. Finally, in Section 7, we draw conclusions and set directions for future work.

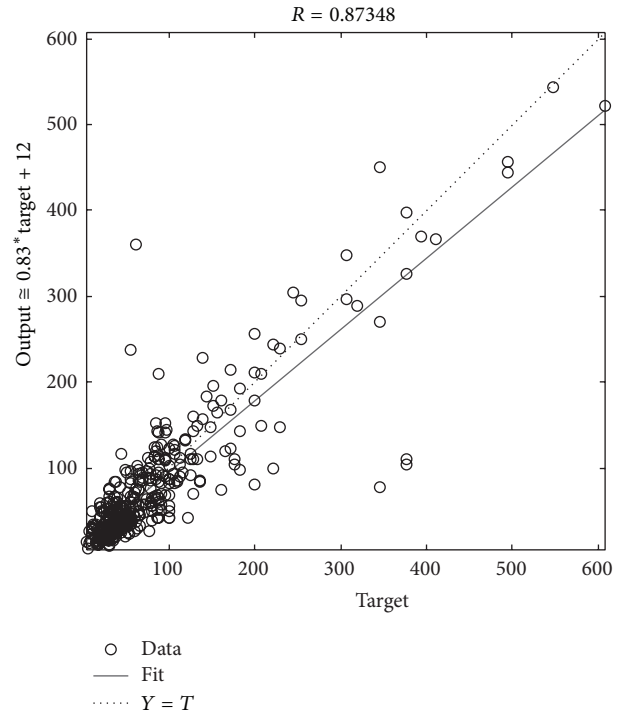


FIGURE 3: GP steel expression no. 2.

2. Problem Formulation

The problem of identifying a corrosion model reduces to defining a function CR that expresses corrosion rate in terms of the n environmental factors that cause it. Such environmental factors will differ from one site to another and include temperature, air humidity, wetness, acidity, concentration of particular chemicals, and so forth. The interaction between the environmental factors and the metal causes corrosion over time. The major influential environmental factors in atmospheric corrosion in the literature are the following.

- (i) Temperature (T): the degree in Celsius; an increase in temperature stimulates corrosion by increasing the rate of electrochemical reactions and diffusion processes [4].
- (ii) Time of wetness (TOW): the time during which the environment's critical relative humidity is greater than 80% and the average temperature is above 0°C ; which forms an electrolyte film on the metal causing its corrosion [4, 8].
- (iii) Sulfur (SO_2): the amount of concentration of the contaminant; sulfur stimulates electrochemical reactions in the electrolyte layer on the metal formed by humidity above 60% to 70% [8].
- (iv) Chloride (Cl^-): the amount of concentration of the contaminant; chloride prevents the creation of protective oxide layers on the metal which accelerates the corrosion process [9].

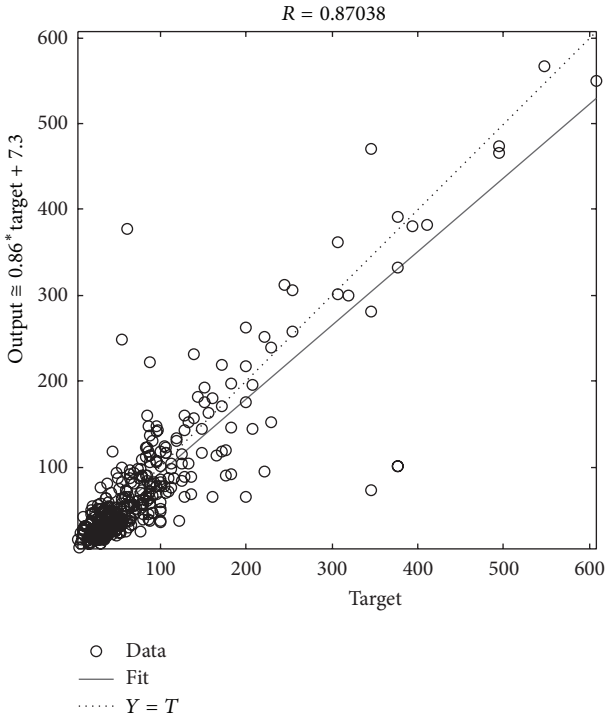


FIGURE 4: GP steel expression no. 3.

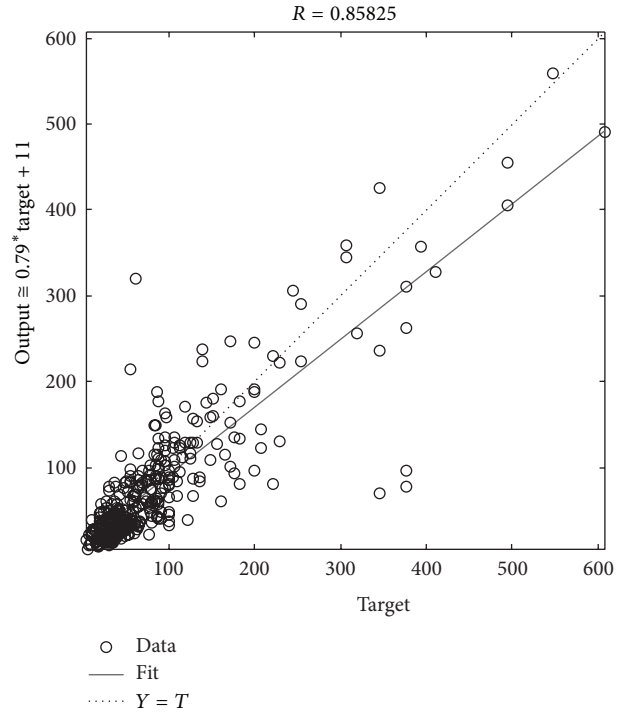


FIGURE 5: GP steel expression no. 4.

- (v) Exposure time (E): the time interval over which the measurements of the previous environmental factors took place.

In our work, the function CR that represents the corrosion rate has five inputs T , TOW , SO_2 , Cl^- , and E that represent the $n = 5$ environmental factors. We shall use the following representation: the environmental factors form an m -by- n matrix X which is input to the function CR and the resulting output corrosion rates form the m -by-1 vector y . Here, m is the number of observations where the values of the environmental factors are recorded together with the corresponding corrosion rates. The corrosion model that we want to identify is thus $CR(X) = y$.

3. Methodology

In order to identify the function CR, we first start by collecting an m -by- $(n + 1)$ matrix of data where m is the number of experiments in which n values of the variables in X have been collected together with the resulting value y . After that, the set of m observations is split into two parts: one for building the model and one for evaluating its accuracy. The part of data that is used for building the model is usually the largest, say $0.9m$ data items, and the part used to evaluate the model is the remaining $0.1m$ data items. This division is not engraved in stone and can be changed while satisfying two opposing criteria: (i) the size of the dataset used in training should be as large as possible to account for a diversity of data points while deriving the expression of interest and (ii) the size of the dataset used in evaluation should be as large as possible to avoid overfitting in the derived model.

After collection of data, we apply the evolutionary technique of interest, that is, genetic programming (Section 3.1) and genetic algorithms (Section 3.2)—in order to determine the function CR.

3.1. Genetic Programming. Genetic programming (GP) [6] is a bioinspired computer algorithm that mimics natural evolution of living organisms. It is similar to genetic algorithms with the exception that individuals are computer programs as opposed to vectors of values.

The objective of GP is to evolve a computer program that solves a given problem. In order to do so, a population of computer programs called individuals—that are randomly generated initially—is evolved across a number of generations. The evolution of the population involves the exchange of genetic material between the individuals through crossover operations and the alteration of the genetic material of single individuals through mutation operations. A selection strategy is applied to the individuals of a population in a given generation to decide which ones are allowed to proceed to the next generation. Such selection is based on the fitness of the individuals which is a problem-dependent value that specifies the goodness of an individual in solving the problem at hand. The evolution continues until a good-enough individual that solves the problem adequately is found, or until a maximum number of generations is reached.

Each individual in the population is a program represented by its abstract-syntax tree (AST). All nonleaf nodes of the AST represent operators, and leaf nodes represent problem variables or constant values. Crossover two programs means taking one or more subtrees from the first program

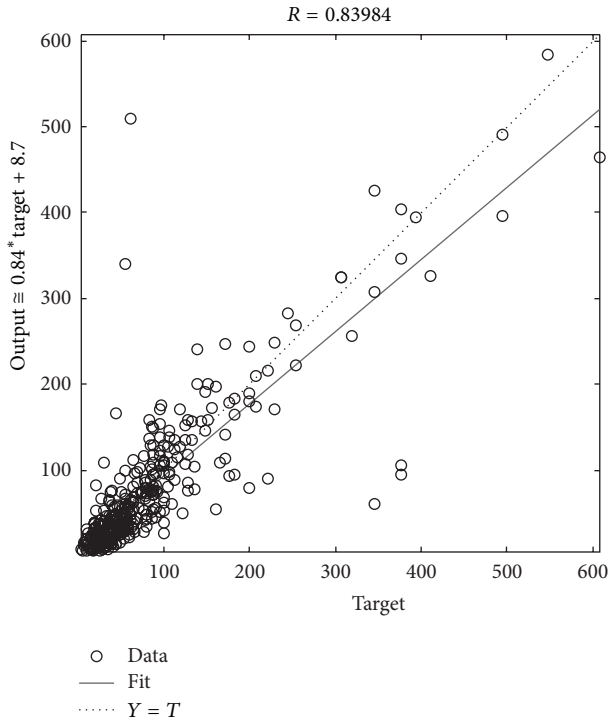


FIGURE 6: GP steel expression no. 5.

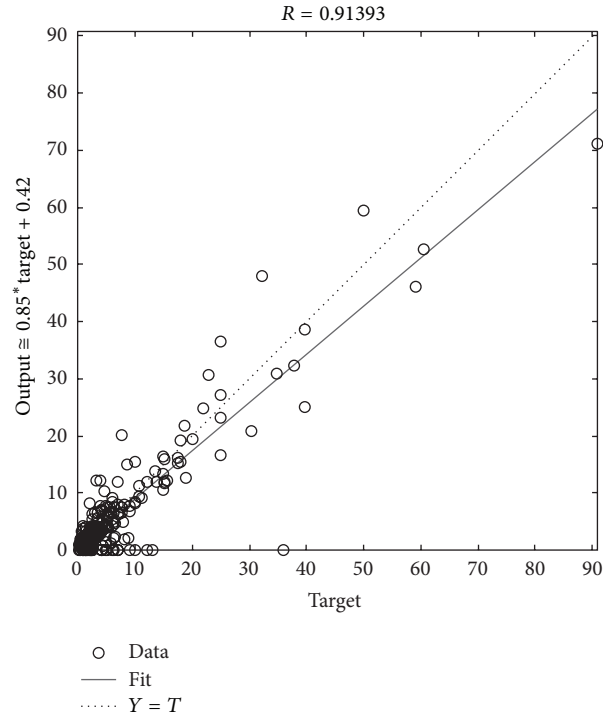


FIGURE 7: GP zinc expression no. 1.

and inserting them into the second program and taking one or more subtrees from the second program and inserting them into the first program (crossovers can be single point or multiple point). Mutating means changing the content of one or more nodes in the AST.

GP can be used to solve a variety of optimization problems amongst which is symbolic regression that we shall describe here because it is the essence of our approach. To solve a symbolic-regression problem (also known as function-discovery problem), the genetic program (GP (we use GP to refer to both “genetic programming” and “genetic program”)) takes as input a set of m observations of values of some variable x , a set of observations of values of some variable y and tries to identify the function f_0 such that $y = f_0(x)$ is true for all pairs (x, y) in the m observations—and also true outside the m observations. The function f_0 to be determined is a computer program that will be evolved by the GP. The initial population of the GP contains a number n of randomly generated functions f_1, f_2, \dots, f_n —each represented as an AST, for example, Figure 1 shows the AST representation of some function $f_i(x) = x^3 + 2x + 5$ in the GP. The functions will be crossed over and mutated over generations to produce new fitter functions. The fitness of a function f_i is calculated as the sum of differences ($|x - y|$) for all pairs (x, y) in the m observations. At the end, the GP either discovers f_0 during evolution or another function of equal or inferior fitness. Notice that the GP can derive some function f'_0 that satisfies $y = f'_0(x)$ for all observed data pairs (x, y) but does not satisfy $y = f'_0(x)$ in the general case, that is, for some unobserved pairs (x, y) the relation $(y = f_0(x) \wedge y \neq f'_0(x))$ holds. This is a classic case of overfitting and

is usually (partially) tackled by dividing the m observations into a training part used during evolution and a testing part used after evolution to give an indication of how well the derived function f'_0 generalizes to new unseen data. Should the derived function not generalize well enough, evolution is restarted with the derived function f'_0 injected in the initial population of the new GP run.

3.2. Genetic Algorithms. Genetic algorithms (GAs) [7] work in a very similar fashion to genetic programming except that each individual during evolution is an array as opposed to a tree. This means that GAs cannot be used to evolve a symbolic expression like GP does since evolving an expression requires the ability to evolve an AST, not a flat array. However, the GA can be used as a powerful regression tool to estimate the coefficients of an expression f whose structure is known already. For example, if we know for example that some function f is $f(x) = a_1x + x^{a_2}$ where x is the independent variable and a_1 and a_2 are constants, we can evolve the array $A = [a_1, a_2]$ such that the difference $f(x) - y$ is minimal in all m observations (as discussed in Section 3.1). The use of GAs in this case is similar to the use of linear and nonlinear regression, however, with the added advantage that it can escape local minima.

4. Evaluation

We use the datasets available from [10] to conduct our experiments. The datasets show the corrosion rates of the two metals: steel and zinc. Corrosion for both metals is measured

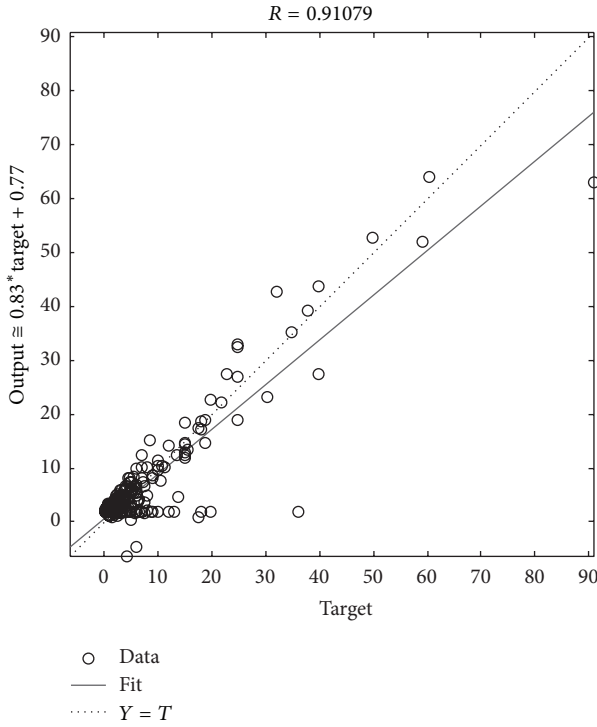


FIGURE 8: GP zinc expression no. 2.

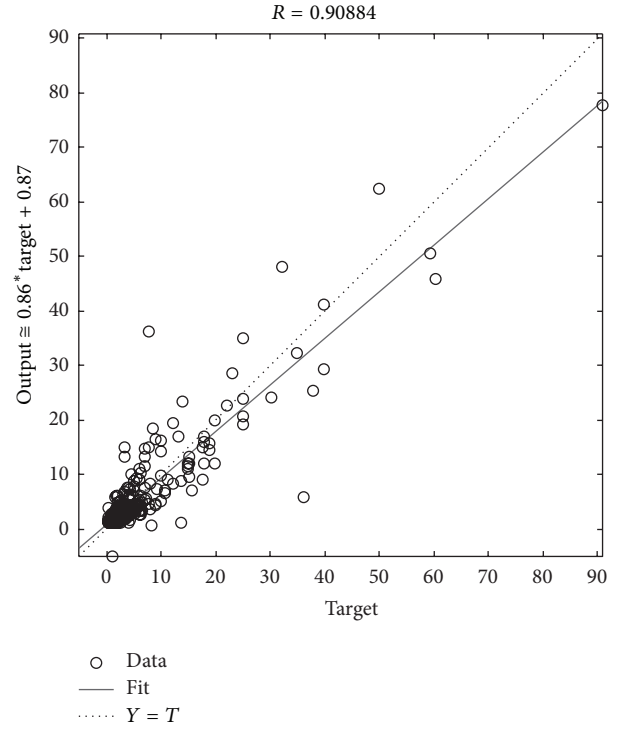


FIGURE 9: GP zinc expression no. 3.

TABLE 1: The steel dataset used in the experiments (adapted from [10, 11]).

	Steel			
	Min	Max	Mean	Sta. Dev.
Temperature (T in $^{\circ}\text{C}$)	-3.03	27.9	11.95	6.67
Time of Wetness (TOW in annual fraction)	0.01	0.95	0.42	0.14
Sulfur (SO_2 in $\text{mg SO}_2 \text{ m}^{-3}$)	0	171	27.31	34.54
Chloride (Cl^- in $\text{mg Cl}^- \text{ m}^{-2} \text{ d}^{-1}$)	0	641	35.81	82.60
Exposure Time (E in years)	0.5	12	2.96	2.56
Corrosion Rate (CR in μm)	5.0	1804.4	83.99	124.32

against the five most influential environmental factors: temperature, time of wetness, concentration of sulfur dioxide, Chloride, and exposure time as explained in Section 2. Tables 1 and 2 show some relevant statistics about the datasets we use in our experiments. In the following we will show the results of applying each evolutionary techniques to determine an expression or model of the corrosion rates of steel and zinc.

4.1. Results of Using GP. The GP was run using the parameters shown in Table 3. The fitness function is aggregate and is computed as the sum of the average mean squared error

TABLE 2: The zinc dataset used in the experiments (adapted from [10, 11]).

	Zinc			
	Min	Max	Mean	Sta. Dev.
Temperature (T in $^{\circ}\text{C}$)	-3.03	27.9	11.65	6.46
Time of Wetness (TOW in annual fraction)	0.01	0.95	0.43	0.14
Sulfur (SO_2 in $\text{mg SO}_2 \text{ m}^{-3}$)	0	12	26.23	32.96
Chloride (Cl^- in $\text{mg Cl}^- \text{ m}^{-2} \text{ d}^{-1}$)	0	125	39.67	90.55
Exposure Time (E in years)	0.5	641	2.79	2.63
Corrosion Rate (CR in μm)	0.2	90.8	5.60	9.56

(MSE) and the complexity of solution measured as the number of nodes in the resulting corrosion expression. The size of the expression was added as penalty to guide the evolution process towards small expressions. The expressions obtained for the corrosion rates of steel and zinc are shown in Tables 4 and 5, respectively—in decreasing goodness-of-fit scores. The value R in Tables 4 and 5 is obtained by performing a regression analysis between the model output (i.e., the corrosion-rate values obtained using the derived expression) and the corresponding target (i.e., the corrosion-rate values available in the dataset). The more R is closer to 1, the more the model is fitting the target data.

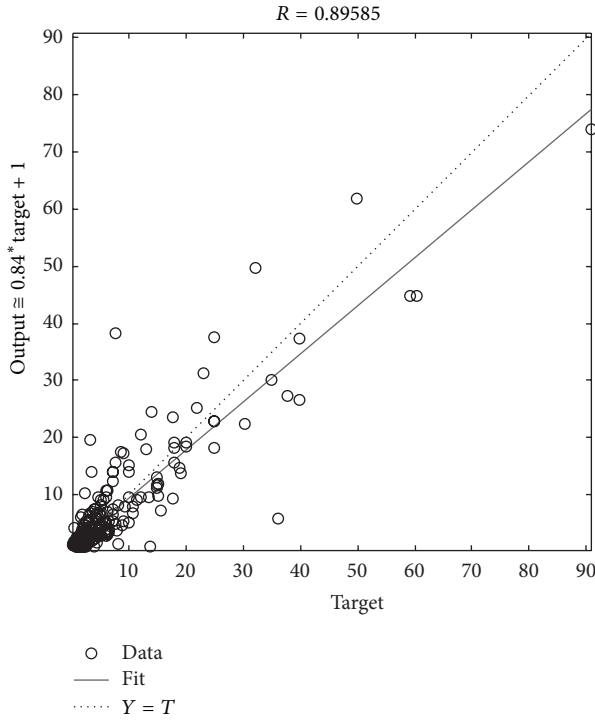


FIGURE 10: GP zinc expression no. 4.

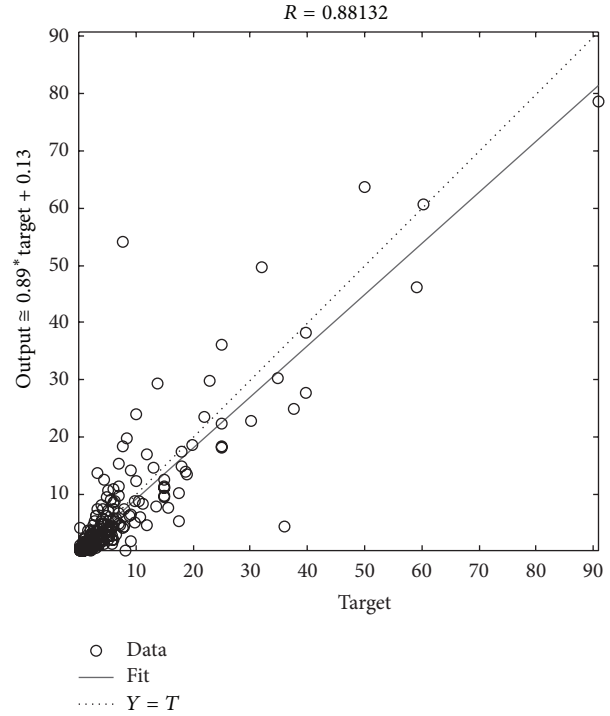


FIGURE 11: GP zinc expression no. 5.

TABLE 3: The GP parameters used in the experiments.

GP Parameter	Value
Population size	100
Maximum number of generations	10,000
Selection strategy	Tournament
Tournament size	10
Maximum initial individual (expression) depth	5
Crossover probability	0.9
Subtree mutation probability	0.1
Number of input variables	5
Number of constants	5
Minimum value for constants	1
Maximum value for constants	5

Figures 2, 3, 4, 5, and 6 show the goodness of fit of the derived GP expressions for steel according to their order in Table 4, and Figures 7, 8, 9, 10, and 11 show the goodness of fit of the derived GP expressions for zinc according to their order in Table 5. In these figures, the variable *Target* on the x -axis shows the measured corrosion rate in the datasets while the variable *Output* on the y -axis shows the estimated corrosion rate using the respective GP expression for the same dataset point. The reported results are obtained by using our own GP implementation. Although there are robust GP systems around such as Eureka Formulize [12], we got the best results using our own GP system especially that we forced all environmental factors to appear in the final symbolic expression—something we had little control over in Eureka Formulize.

As can be seen, the GP expressions have very high goodness-of-fit values despite the necessarily noisy datasets. The GP ran for around 60 minutes to derive the corrosion-rate expressions of each metal before reaching the maximum number of generations.

4.2. Results of Using GAs. We used the robust MATLAB *ga* library to obtain the best results. We used the corrosion expression (1) from [13] where the constants $a_i, i \in [1 \dots 10]$ are the ones to evolve using the GA. Table 6 lists the GA parameters used in the evaluation:

$$\text{CR} = a_1 \times E^{a_2} \left(\frac{\text{TOW}}{a_3} \right)^{a_4} \times \left(1 + \frac{\text{SO}_2}{a_5} \right)^{a_6} \times \left(1 + \frac{\text{Cl}^-}{a_7} \right)^{a_8} \times e^{a_9(T+a_{10})}. \quad (1)$$

Initially, the GA yielded very inaccurate expressions for steel and zinc (the error was in the range 10^{18})—which was rather unexpected. However, a closer investigation revealed that the GA was not performing protected division that is, during the evaluation of the fitness of an individual that has $a_3 = 0$, $a_5 = 0$, or $a_7 = 0$ the fitness values were erroneous.

To circumvent this problem, we penalized individuals that have $a_3 = 0$, $a_5 = 0$, or $a_7 = 0$ by assigning poor fitness values to them. The GA corrosion expressions for steel and zinc are shown in (2) and (3), respectively, with MSE values

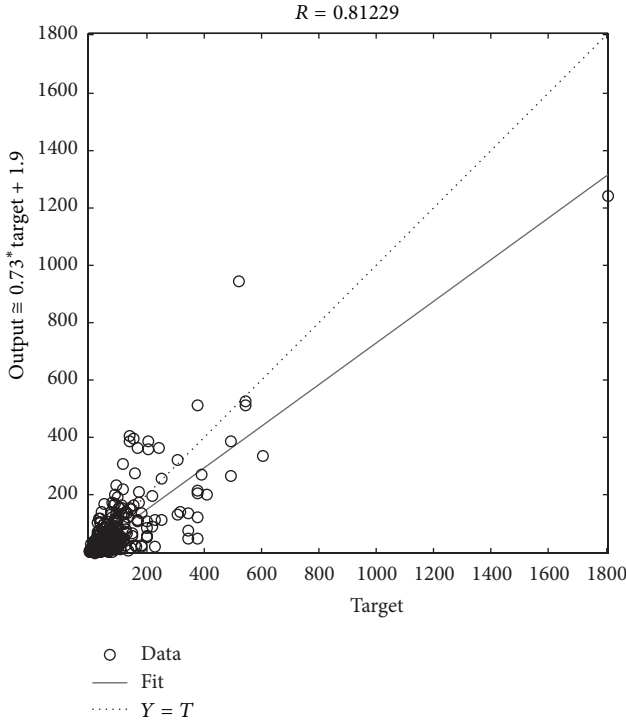


FIGURE 12: GA steel expression.

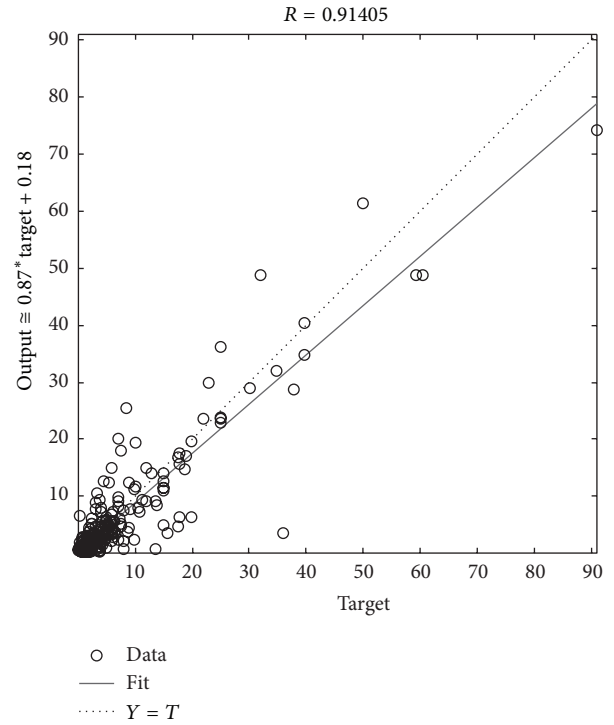


FIGURE 13: GA zinc expression.

of 5791 and 15.9, respectively. The goodness-of-fit plots are shown in Figure 12 for steel and Figure 13 for zinc:

$$\begin{aligned} \text{CR} = & 1.5124E^{1.2406} \times \left(\frac{\text{TOW}}{0.1300} \right)^{0.8967} \\ & \times \left(1 + \frac{\text{SO}_2}{0.0002} \right)^{-0.0813} \times \left(1 + \frac{\text{Cl}^-}{1.0406} \right)^{0.3787} \\ & \times e^{0.0650(T+1.4088)}, \end{aligned} \quad (2)$$

$$\begin{aligned} \text{CR} = & 0.1757E^{1.0376} \times \left(\frac{\text{TOW}}{19.4277} \right)^{-0.1107} \\ & \times \left(1 + \frac{\text{SO}_2}{5.4385} \right)^{0.3866} \times \left(1 + \frac{\text{Cl}^-}{9.4377} \right)^{0.5633} \\ & \times e^{0.0426(T-4.1541)}. \end{aligned} \quad (3)$$

As can be seen, the GA expressions are also accurate. The GA ran for around 15 minutes to derive each of the reported expressions before reaching the maximum number of generations.

5. Discussion

Table 7 shows a summary of the accuracy of the two evolutionary techniques for predicting corrosion rates for steel and zinc.

In terms of usefulness, the GP expressions are superior to the GA expressions because they are derived automatically, that is, without knowledge about the structure of

the target corrosion-rate expression, whereas we assume a specific corrosion-expression structure for GAs and evolve its parameters. The explicit GP corrosion-rate expressions give more insight into the corrosion process because they show how corrosion rate is affected by the environmental factors.

The datasets used in the experiments are characterized by the presence of a number of outliers which can either be (i) genuine data points where corrosion rate deviates significantly from average because of the inherent complexity of the corrosion process or (ii) erroneous data points that result from faults in measurement devices, human mistakes during data entry, and so forth. The analysis presented in this paper assumes case (i), that is, all data points are assumed to be valid.

As can be seen from the results, the slope of the fitting line is seemingly controlled by outliers. In order to investigate this issue further, we have redone the GP evolution, however this time by a better handling of the outliers using two methods as follows. First, we used a logarithmic distance measure $\log(1 + |y - f(x)|)$ instead of the squared error $(y - f(x))^2$. The outliers in the dataset are data points of large magnitudes, which means that if they do not lie close to the curve during evolution, they will affect the fitness of the solution in a substantial way if their distance from the curve is measured by, for example, $(y - f(x))^2$. When the distance is logarithmic, the effect of outliers on the evolution of the curve is significantly reduced, for example, if $(y - f(x))^2 = 10^6$ then $\log(1 + |y - f(x)|) \approx 3$. Second, we removed the outliers all together and used the squared error $(y - f(x))^2$ as we did previously.

TABLE 4: The best five expressions obtained for the corrosion expression of steel using the GP. The fitness of an expression is its mean squared error (MSE), and its size is the number of nodes in its underlying abstract-syntax tree.

Rank	Size	MSE	R	Expression
1	31	788	0.87523	$T + \frac{41E}{10} + \frac{63TOW Cl^-}{100} + \frac{7E^2}{25} + SO_2 \left(\frac{63TOW}{100} + \frac{7E}{25} \right) + \frac{7T \cdot TOW E^2}{25SO_2}$
2	41	810	0.87348	$T + \frac{11E}{5} + \frac{31E^2}{10SO_2} + \frac{8E(SO_2 + (11/5))}{25} + \frac{69TOW(T + SO_2 + (83Cl^-/100))}{100} - \frac{6}{5}$
3	27	831	0.87038	$T + \frac{3TOW(SO_2 + Cl^-)}{5} + \frac{3E((3SO_2/5) + (3TE/5SO_2) + (13/5))}{5} + \frac{17}{10}$
4	49	1036	0.85825	$T + SO_2 - \frac{2SO_2 - Cl^-}{E - TOW + 3} + E^2 + \frac{23TOW SO_2 E}{50} + \frac{23TOW((37SO_2/10) - (37T/10) + 10TOW E(T - (13/10)))}{100SO_2}$
5	69	878	0.83984	$\frac{57SO_2 - E}{100} - E + \frac{7TOW(TOW + (23Cl^-/50) - ((SO_2 - (23SO_2 E/5))/T))}{(E + T \cdot TOW)} + \frac{(6E/25) + (6E((SO_2 - (23SO_2 E/5))/T - 3SO_2 + (21/5)))/25}{SO_2}$

TABLE 5: The best five expressions obtained for the corrosion expression of zinc using the GP.

Rank	Exp. size	Exp. fitness	R	Expression
1	41	13	0.91393	$\frac{E \left(\text{SO}_2 - T + \left(\text{Cl}^- / \text{TOW} - \text{SO}_2 + (T / \text{TOW}) \right) + (T / \text{TOW}) + (\text{TOW}^2 \text{Cl}^- / E) \right)}{(T / \text{TOW}) + (\text{TOW} \text{SO}_2 / \text{Cl}^-) + 1}$
2	49	14	0.91079	$\frac{\text{SO}_2 E}{(8 T / 5) + \text{TOW} + \left(\text{SO}_2 + (9 / 5) / \text{Cl}^{-2} \right) - (63 \text{Cl}^- ((63 T / 100) - (8 / 5)) / 100 \text{SO}_2) + (41 / 5)} + \frac{9}{5}$
3	35	13	0.90884	$\frac{37 E ((13 \text{SO}_2 / 20) - (8 \text{TOW} / 25) + (17 \text{Cl}^- / 25) + (8 \text{Cl}^- ((11 E / 10) - 10) / 25 T))}{1000} + \left(\frac{6}{5} \right)$
4	27	15	0.89585	$\frac{13 E}{250} - \frac{7 \text{TOW}}{20} + \frac{\text{SO}_2 E}{40} + \frac{41 \text{TOW} \text{Cl}^- E}{1000} + \frac{11}{10}$
5	31	18	0.88132	$\frac{3 \text{TOW} \text{SO}_2 E}{10 (2 \text{TOW} + (14 / 5))} + \frac{\text{TOW}^2 \text{Cl}^- E^2}{4 \text{SO}_2}$

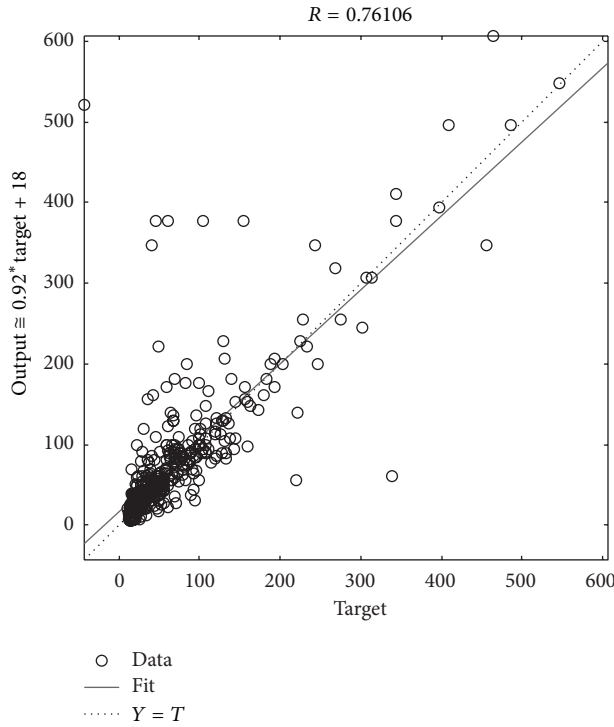


FIGURE 14: GP steel expression using logarithmic error.

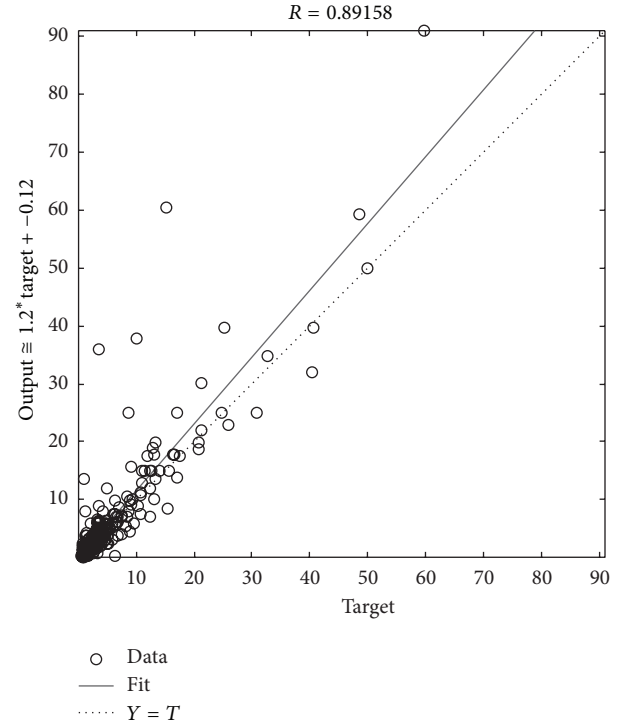


FIGURE 15: Zinc expression using logarithmic error.

TABLE 6: The GA parameters used in the experiments.

GA Parameter	Value
Maximum number of generations	10,000
Size of individual	10
Population size	20 ^(*)
Selection strategy	Stochastic uniform ^(*)
Crossover function	Scattered Points ^(*)
Mutation function	Gaussian ^(*)

^(*)Indicates MATLAB default.

TABLE 7: Summary of Model-Identification Goodness of Fit (values R).

Evolutionary technique	R of steel	R of zinc
Genetic programming	0.87523	0.91393
Genetic algorithms	0.81229	0.91405

The resulting corrosion-rate expression for steel using the logarithmic error is shown in (4); the resulting corrosion-rate expression for zinc using the logarithmic error is shown in (5) the resulting corrosion-rate expression for steel after dropping outliers is shown in (6), and the resulting corrosion-rate expression for zinc after dropping outliers is shown in (7). Figures 14, 15, 16, and 17 show the goodness of

fit of the corrosion-rate expressions (4), (5), (6), and (7), respectively:

$$\begin{aligned} \text{CR} = & 12.05 \times \text{TOW} + 11.05 \times E + 0.1619 \times \text{Cl}^- \\ & + 0.772 \times \text{TOW} \times \text{SO}_2 \times E - 0.772 \times \text{TOW}^3 \times E^3, \end{aligned} \quad (4)$$

$$\begin{aligned} \text{CR} = & 0.6053 + \text{TOW} \times E + 0.03341 \times \text{TOW}^2 \times \text{Cl}^- \\ & + (\text{TOW} + 2.006 \times \text{SO}_2 \times E + \text{SO}_2 \times \text{Cl}^- \times E \\ & - 2.006 - \text{SO}_2 - \text{Cl}^-) \\ & \times (172.8 + 2.309 \times T \times \text{Cl}^-)^{-1}, \end{aligned} \quad (5)$$

$$\begin{aligned} \text{CR} = & \text{SO}_2 + 24.04 \times \text{TOW} + 8.566 \times E + 0.3724 \times \text{Cl}^- \\ & - \frac{0.7507 \times \text{SO}_2}{E}, \end{aligned} \quad (6)$$

$$\begin{aligned} \text{CR} = & \text{TOW} + 0.6582 \times E \\ & + (0.004668 \times T \times \text{SO}_2 \times E + 0.004337 \times \text{SO}_2 \\ & \times \text{Cl}^- \times E) \times (T + \text{TOW} - 2.304)^{-1}. \end{aligned} \quad (7)$$

As can be seen from the results, using GP still gave good results when outliers were eliminated and when their effect was significantly reduced.

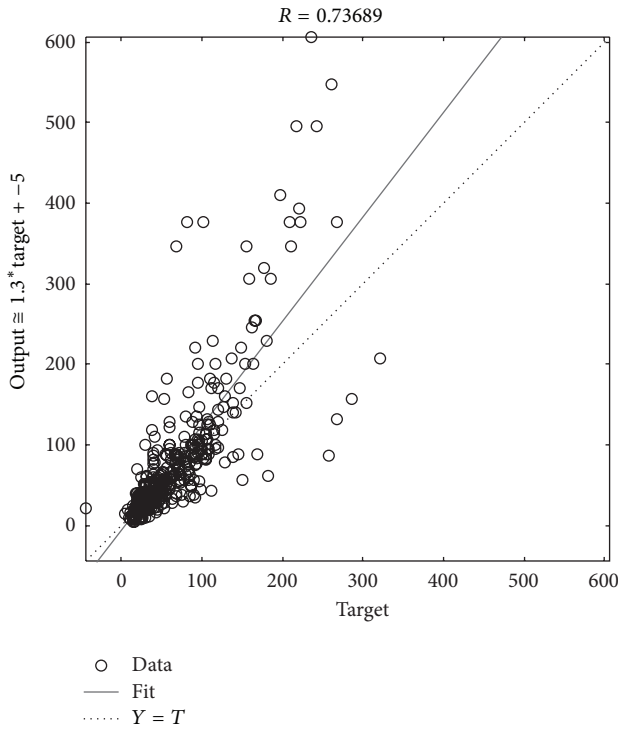


FIGURE 16: GP steel expression after dropping outliers.

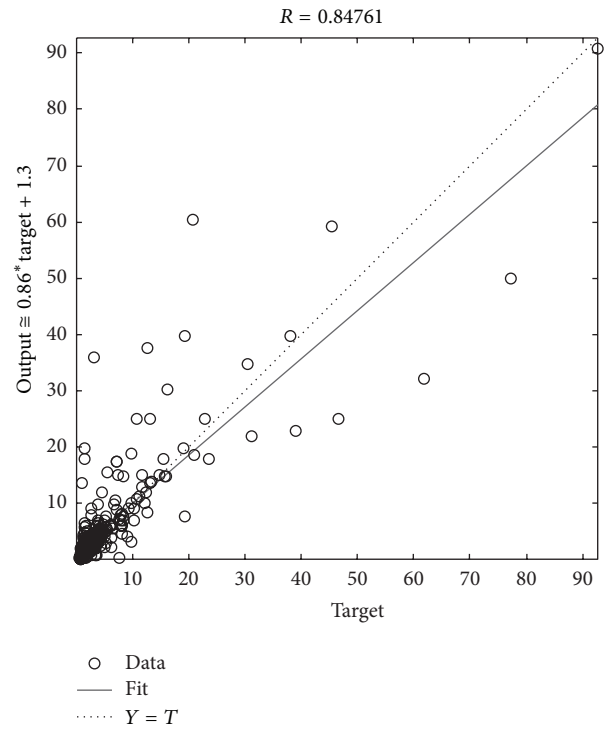


FIGURE 17: GP zinc expression after dropping outliers.

TABLE 8: Corrosion models in the literature.

Expression	Citation
$CR = 0.16TOW^{0.7}(SO_2 + 1.78)$	[14]
$CR = 325E^{0.5}e^{0.00275SO_2 - (163.2/RH)}$	[15]
$CR = 2.28SO_2 + 176.6$	[16]
$CR = 18.5SO_2 + 292.5$	[16]
$CR = 0.0152TOW^{0.428}SO_2^{0.570}$	[17]
$CR = 1.17TOW^{0.66}(SO_2 + 0.048)$	[18]
$CR = 4.0SO_2 + 58$	[19]
$CR = 8.0SO_2 + 160$	[19]
$CR = 1.327 + 0.4313SO_2 + 0.0057TOW + 0.1384Cl^-$	[19]
$CR =$ $a_1 \cdot E^{a_2} \left(\frac{TOW}{a_3} \right)^{a_4} \cdot \left(1 + \frac{SO_2}{a_5} \right)^{a_6} \cdot \left(1 + \frac{Cl^-}{a_7} \right)^{a_8} \cdot e^{a_9(T+a_{10})}$ [13] $a_3 \in \{3300, 4300\}$, $a_4 \in \{0.02, 0.23, 0.24, 0.46\}$, $a_5 = 25$, $a_7 = 50$, $a_{10} = 20$. a_1, a_2, a_6, a_8, a_9 derived using least-square method.	

RH is relative humidity.

6. Related Work

Corrosion modeling is not a novel research area. Many corrosion models have been developed in the literature to yield expressions of metal corrosion as shown in Table 8. Most of these models do not take into account all five environmental factors that we consider in this work (see Table 8).

In addition to this, corrosion modeling has also been attempted using artificial neural networks in numerous

works including [10, 20–22] and using support vector regression [11]; however, these techniques do not yield explicit corrosion-rate expressions.

7. Conclusions

In this paper, we have developed a corrosion model based on two evolutionary computation techniques, namely, genetic programming and genetic algorithms. Both techniques yielded corrosion-rate expressions with good accuracy with genetic programming being superior because it can learn without prior knowledge the structure of the corrosion expression. The findings of this work will allow better understanding of the corrosion phenomenon in terms of cause and effects so that necessary action such as prevention measures can be carried out.

Acknowledgment

This research is funded by the Institute of Consulting Research & studies, Umm Al-Qura University, Makka, Saudi Arabia, Grant no. S2011-2.

References

- [1] CC Technologies Laboratories and NACE International, "Corrosion costs and preventive strategies in the United States," Publication FHWARD- 01-156, U.S. Federal Highway Administration, Washington, DC, USA, 2001.
- [2] R. W. Revie, *Uhlig's Corrosion Handbook*, CANMET Materials Technology Laboratory; John Wiley & Sons, Ontario, Canada, 2000.

- [3] NACE International, *Glossary of Corrosion Related Terms*, NACE International, Houston, Tex, USA, 2002.
- [4] M. G. Fontana, *Corrosion Engineering*, McGraw-Hill, New York, NY, USA, 3rd edition, 1986.
- [5] Saudi Basic Industries Corporation, "SABIC," <http://www.sabic.com/>.
- [6] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, The MIT Press, Cambridge, Mass, USA, 1992.
- [7] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, Mass, USA, 1989.
- [8] P. R. Roberdge, *Handbook of Corrosion Engineering*, McGraw-Hill, New York, NY, USA, 2000.
- [9] G. G. Naumov, B. N. Ryzenko, and I. L. Khodakovsky, *Handbook of Thermodynamic Data*, 1974, USGS Translation, USGS-WRD-74-000.
- [10] J. Cai, R. A. Cottis, and S. B. Lyon, "Phenomenological modelling of atmospheric corrosion using an artificial neural network," *Corrosion Science*, vol. 41, no. 10, pp. 2001–2030, 1999.
- [11] S. F. Fang, M. P. Wang, W. H. Qi, and F. Zheng, "Hybrid genetic algorithms and support vector regression in forecasting atmospheric corrosion of metallic materials," *Journal of Computational Materials Science*, vol. 44, no. 2, pp. 647–655, 2008.
- [12] Cornell Creative Machines Lab, "Eureqa," <http://creativemachines.cornell.edu/eureqa>.
- [13] D. E. Klinesmith, R. H. McCuen, and P. Albrecht, "Effect of environmental conditions on corrosion rates," *Journal of Materials in Civil Engineering*, vol. 19, no. 2, pp. 121–129, 2007.
- [14] H. Guttman and P. J. Sereda, *Metal Corrosion in the Atmosphere*, vol. 558 of *ASTM Special Technical Publication*, ASTM, Philadelphia, Pa, USA, 1968.
- [15] F. H. Haynie and J. B. Upham, "Correlation between corrosion behavior of steel and atmospheric pollution data," in *Corrosion in Natural Environments*, vol. 558 of *ASTM Special Technical Publication*, pp. 33–51, ASTM, Philadelphia, Pa, USA, 1974.
- [16] L. Atteraas, S.E. Haagenrud, and V. Kucera, "Corrosion of steel and zinc in Scandinavia with respect to the classification of the corrosivity of atmospheres," in *Proceedings of the 8th Scandinavian Corrosion Congress*, Helsinki University of Technology, Helsinki, Finland, 1978.
- [17] K. Barton, "Schutz gegen atmosphärische Korrosion, Theorie und Technik, Verlag Chemie," *Protection of Metals*, vol. 16, p. 387, 1980.
- [18] T. Hakkarainen and S. Yläsaari, "Atmospheric corrosion testing in Finland," in *Atmospheric Corrosion*, W. H. Ailor, Ed., pp. 787–795, John Wiley & Sons, New York, NY, USA, 1982.
- [19] D. Knotkova, J. Gullman, P. Holler, and V. Kucera, "Assessment of corrosivity by short-term atmospheric-field tests for technically-important metals," in *Proceedings of the 9th International Congress on Metallic Corrosion*, National Research Council, Toronto, Canada, 1984.
- [20] S. Hernández, S. Nešić, G. Weckman, and V. Ghai, "Use of artificial neural networks for predicting crude oil effect on carbon dioxide corrosion of carbon steels," *Corrosion*, vol. 62, no. 6, pp. 467–482, 2006.
- [21] W. You and Y. Liu, "Predicting the corrosion rates of steels in sea water using artificial neural network," in *Proceedings of the 4th International Conference on Natural Computation (ICNC '08)*, vol. 1, pp. 101–105, October 2008.
- [22] S. Tesfamariam and B. Martín-Pérez, "Bayesian belief network to assess carbonation-induced corrosion in reinforced concrete," *Journal of Materials in Civil Engineering*, vol. 20, no. 11, pp. 707–717, 2008.

