Neutrosophic rule-based prediction system for toxicity effects assessment of biotransformed hepatic drugs

Sameh H. Basha\textsuperscript{a}, Alaa Tharwat\textsuperscript{b,d,1,}, Areeg Abdalla\textsuperscript{a}, Aboul Ella Hassanien\textsuperscript{c,d}

\textsuperscript{a}Faculty of Science, Cairo University, Egypt
\textsuperscript{b}Faculty of Computer Science and Engineering, Frankfurt University of Applied Sciences, Frankfurt am Main, Germany
\textsuperscript{c}Faculty of Computers and Information, Cairo University, Egypt
\textsuperscript{d}Scientific Research Group in Egypt (SRGE), Egypt

A R T I C L E   I N F O

Article history:
Received 27 April 2018
Revised 7 December 2018
Accepted 8 December 2018
Available online 11 December 2018

Keywords:
Toxic effects
Drugs design
Neutrosophic theory
Rule-based classification
Prediction model

A B S T R A C T

Measuring toxicity is an important step in drug development. However, the current experimental methods which are used to estimate the drug toxicity are expensive and need high computational efforts. Therefore, these methods are not suitable for large-scale evaluation of drug toxicity. As a consequence, there is a high demand to implement computational models that can predict drug toxicity risks. In this paper, we used a dataset that consists of 553 drugs that biotransformed in the liver. In this data, there are four toxic effects, namely, mutagenic, tumorigenic, irritant and reproductive effects. Each drug is represented by 31 chemical descriptors. This paper proposes two models for predicting drug toxicity risks. The proposed models consist of three phases. In the first phase, the most discriminative features are selected using rough set-based methods to reduce the classification time and improve the classification performance. In the second phase, three different sampling algorithms, namely, Random Under-Sampling, Random Over-Sampling, and Synthetic Minority Oversampling Technique (SMOTE) are used to obtain balanced data. In the third phase, the first proposed model employs the Neutrosophic Rule-based Classification System (NRCS), and the second model uses Genetic NRCS (GNRCS) to classify an unknown drug into toxic or non-toxic. The experimental results proved that the proposed models obtained high sensitivity (89–93\%), specificity (91–97\%), and GM (90–94\%) for all toxic effects. Overall, the results of the proposed models indicate that it could be utilized for the prediction of drug toxicity in the early stages of drug development.

© 2018 Elsevier Ltd. All rights reserved.

1. Introduction

The development of new drugs is an expensive and complex process, and it has many steps (Pereira et al., 2009). One of the main steps in this process is the toxicity assessment of drugs' components. This step is vital as it is used for predicting drug failures before any clinical trials. Therefore, this step could save 100 million dollars per one drug development which reflects the importance of measuring toxicological effects as early as possible (Pritchard et al., 2003; Ulrich & Friend, 2002). Hence, measuring toxicity for thousands of compounds becomes a hot topic in the recent studies (Huang et al., 2009; von Korff & Sander, 2006).

Toxicity of a drug refers to the undesirable effects of the drug on the whole organism (e.g. animal or plant), an organ (e.g. liver), or substructure of the organism (e.g. a cell). However, reliable high-throughput assays are expensive and time-consuming; thus, there is a high demand for computational models. The computational models are faster and cheaper alternatives to in-vivo and in-vitro bioassays. Also, they save experimental materials and protect animals. Thereby, utilizing the computational models enables the pharmaceutical industry to produce over 100,000 new drugs yearly and save animal trials as well (Cao et al., 2012; Plewczynski, 2008; Tharwat, Gaber, Fouad, Snasel, & Hassanien, 2015). These computational models have different goals such as predicting the toxicity of the chemical compounds, estimating the effect of different concentrations of the chemical compounds or predicting the toxicological endpoints.

There are many examples of available computer models predicting toxicity such as OnkoLogic (Woo, Lai, Argus, & Arcos, 1995), TOPKAT (Privat, 2001), DEREK (Woo et al., 1995), Case (Klopman, 1984) and Multicase (Klopman, 1992). Tharwat, Gabel,
and Hassani (2017), Tharwat, Moemen, and Hassanian (2017) introduced two models for predicting the toxicity of drugs, on the same dataset used in this paper. They used the Whale Optimization and Dragonfly algorithms for finding the optimal parameters of the Support Vector Machine (SVM) classifier, and their model obtained competitive results. However, they reported that due to the stochastic nature of the Whale optimization and the Dragonfly algorithms, there is no guarantee for finding the optimal solution. Moreover, there are many factors that may affect the quality of their models such as the representativeness and the diversity of training data. Multiclassification systems (MCSs) (also called multiclassifiers or classifier ensembles) are rule-based systems that combine a set of different classifiers by assigning weights to each classifier individual decision. There have been different strategies developed to ensemble classifiers with accuracy-complexity trade-offs. Fuzzy Unordered Rules Induction Algorithm (FURIA), Bagging FURIA-based Fuzzy are two well-known examples of the rule-based systems (Trawiński, Cordón, & Quirin, 2011).

In this paper, two proposed classification models are introduced, namely, Neutrosophic Rule-based Classification System (NRCS) and Genetic NRCS (GNRCS). The proposed models were constructed based on the neutrosophic set theory for handling incomplete as well as inconsistent information. Hence, our proposed model can solve many real problems such as missed data. Additionally, in contrast to the models that were proposed in Tharwat, Gabel et al. (2017), Tharwat, Moemen et al. (2017), the NRCS algorithm is deterministic and there is no need to tune any parameters.

One of the main problems of computational toxicity models is the large amount of data. This makes the analysis of this data more difficult because not all the information is relevant. Selecting the relevant information is an important step in prediction models. Feature selection techniques are used to find a subset of features that improves the classification performance and provide a faster classification. The dataset in this research has 31 features, and the first goal of our model is to select the most discriminative features. In the proposed model, three different rough set-based algorithms were used for feature selection.

Another important problem in real applications is the imbalanced data. This problem results from the biased distribution of different classes. In other words, one class has more samples than the other class(es). Hence, the prediction model will not have enough minority samples to train the model. As a consequence, minority samples tend to be misclassified. In the proposed model, the current dataset is imbalanced (see Section 2), and different algorithms were introduced for solving the imbalanced data problem. Moreover, in our experiments, three sampling algorithms were used to obtain balanced classes.

This paper proposes a novel model to evaluate the toxicity of hepatic drugs. The toxicity risks of the current drugs include mutagenic, tumorigenic, irritant and reproductive effects. Each drug is represented by 31 features and there are four class labels, one class label for each toxic effect. For example, there is a class label which indicates if the current drug has the mutagenic effect or not. The classification step is an important step in the prediction model. In this step, a classifier is used for classifying the testing or unseen data to toxic or non-toxic.

There are many traditional classifiers that are used in this area of research. In the proposed model, NRCS is used for classifying the testing data. In this step, the NRCS is used for extracting information from data and then generating rules for the training and testing data. Each testing rule is to be matched with all training rules and the closest class label of the training rule is assigned to the testing rule. The NRCS model is, then, modified using Genetic Algorithms (GA), and the proposed model is called GNRCS. In GNRCS, the GA is used for selecting the most discriminative rules, removing all redundant rules, and generating new rules for exploring the input space. This step increases the accuracy of the proposed model and reduces the number of rules and hence reduces the required computational time.

The rest of this paper is organized as follows: Section 2 presents a brief description of the dataset that is used in our proposed models. Theoretical background and the steps of the proposed models are presented in Section 3. Experimental scenarios and discussions are introduced in Section 4. Finally, conclusions and future work are presented in Section 5.

2. Description of the dataset

In this research, the dataset was extracted from the Drug bank database, which has 6712 drugs. These drugs are classified as follows: 1448 FDA-approved small molecule drugs, 131 FDA-approved biotech (protein/peptide) drugs, 85 nutraceuticals, and 5080 experimental drugs (Sander, Freyss, von Korff, & Rufener, 2015). In our experiments, 553 drugs that are biotransformed in the liver are used; all these drugs are approved by the FDA (Food and Drug Administration). Table 1 shows details about the dataset. As shown, each drug is represented by 31 features, and these features were calculated using DataWarrior package (Sander et al., 2015). The dataset has four toxic effects as indicated in Table 2. The table shows the number of samples in each class, positive and negative samples. As shown, the data is not balanced and the imbalance ratio (the ratio of the number of samples of the majority class to the number of samples of the minority class) is different for the four classes. Moreover, the mutagenic, irritant and tumorigenic effects have a high imbalance ratio compared with the reproductive effect which has a low imbalance ratio. In the data, the positive class represents the minor class. This may have a negative influence on the sensitivity measure of the proposed models. Additionally, the reproductive effect has the top risk effect (33.82%); and the risk effect of the mutagenic and tumorigenic effects are equal (16.28%), and finally the irritant effect has (12.16%) for the current FDA drugs, which reflects the burden on the liver and such drugs should be replaced with safer medications. It is worth mentioning here, in this research, we have four class labels; one class label for each toxic effect.

3. Theoretical background

3.1. Neutrosophic set and neutrosophic logic

Neutrosophic was introduced by Smarandache in 1995, it deals with the origin, nature, and scope of neutralities, as well as their interactions with different mental visions (Wang, Smarandache, Sunderraman, & Zhang, 2005). This theory considers three concepts: (1) the idea \(<A>\), (2) its opposite \(<\text{Anti–A}>\), and (3) a spectrum of “neutralities” \(<\text{Neut–A}>\). The \(<\text{Neut–A}>\) and \(<\text{Anti–A}>\) together are referred to as \(<\text{Non–A}>\) (Wang et al., 2005). Both \(<\text{Anti–A}>\) and \(<\text{Non–A}>\) are used to neutralize and balance the idea \(<A>\) (Basha, Abdalla, & Hassanien, 2016a; Wang et al., 2005).

The fuzzy set (FS) theory was introduced by A. L. Zadeh in 1965 to handle vague and fuzzy information. The FS is represented by a membership degree say \(\mu_A(x)\) for each element \(x\) in a set \(A\), where \(\mu_A(x) \in [0, 1]\) (Zadeh, 1996). Hence, instead of having a class label for each sample, in the fuzzy rule-based system, a membership function maps each sample to a membership value between zero and one to represent its degree of belongingness to a class. Mathematically, given \(C\) classes, any sample \(x\) has a degree \(\mu(x)\) that determines its membership value or score for each class \(c\) (Amons, Montero, Biging, & Cutillo, 2004; Ansari, Biswas, & Aggarwal, 2013). For example, let \(C = 3\), i.e. three classes, and...
$\mu(x) = \{0.2, 0.5, 0.3\}$. This means that the membership value of the second class is larger than the other two classes for $x$.

The fuzzy set theory was generalized by many theories such as interval-valued fuzzy sets (Turksen, 1986), intuitionistic fuzzy sets (Atanassov, 1989), and interval-valued intuitionistic fuzzy set (Atanassov, 1989). Each of these theories can handle only one aspect of imprecision. The FS theory cannot deal with incomplete and inconsistent information. For this reason, the neutrosophic set was constructed to handle incomplete as well as inconsistent information. Moreover, the neutrosophic set is a huge formal structure which generalizes all of the sets such as, the classic set, fuzzy set, interval-valued fuzzy set, intuitionistic fuzzy set, and interval-valued intuitionistic fuzzy set (Arora, Biswas, & Pandy, 2011).

3.11. Neutrosophic set

The fundamental concepts of neutrosophic sets were introduced by Smarandache (2003) and Alblowi, Salama, and Eisa (2013). They provided a natural foundation for treating mathematically the neutrosophic phenomena for building new branches in neutrosophic mathematics.

Mathematically, an element $x(t, i, f)$ belongs to a neutrosophic set $A$, in the following way: $t$ true degree in $A$, $i$ indeterminate degree in $A$ and $f$ false degree in $A$, where $t$, $i$, and $f$ are real numbers taken from the sets $T$, $I$, and $F$, respectively, with no restriction on $t$, $i$, and $f$, nor on their sum $n = t + i + f$ (Smarandache, 2003). For example, assume $x(0.5, 0.2, 0.4)$ belongs to $A$. This means that $x$ is in $A$ with $0.5$%, not in $A$ with $0.4$% and $0.2$% indeterminacy degree. In another example, $y(0, 0, 1)$ belongs to $B$ means $y$ is 100% not in $B$.

Let $X$ be a space of points (objects) with a generic element in $X$ denoted by $x$. A neutrosophic set $A$ in $X$ is characterized by a truth-membership function $(T_A)$, an indeterminacy-membership function $(I_A)$, and a falsity membership function $(F_A)$. The three functions $T_A$, $I_A$, and $F_A$ are real standard or non-standard subsets of $\{0, 1+\}$

\[ T_A = \{1+\} - T_A, \quad I_A = \{1+\} - I_A, \quad F_A = \{1+\} - F_A. \]

That is, $T_A : X \rightarrow [0, 1+]^*$, $I_A : X \rightarrow [0, 1+]^*$, and $F_A : X \rightarrow [0, 1+]^*$. There is no restriction on the sum of $T_A(x)$, $I_A(x)$ and $F_A(x)$, so, $\sum_0^1 \sup T_A(x) + \sup I_A(x) + \sup F_A(x) \leq 3^*$, where $\sup T_A$, $\sup I_A$, and $\sup F_A$ represent the supremum of the $T_A$, $I_A$ and $F_A$, respectively, (Ansari et al., 2013).

There are different ways to construct neutrosophic set operators (Smarandache, 2003).

- **Complement**: The complement of a neutrosophic set $A$ denoted by $A$ and defined by Smarandache (2003), Wang et al. (2005), Arora et al. (2011):

\[ T_b(x) = \{1+\} - T_A(x), \quad I_b(x) = \{1+\} - I_A(x), \quad F_b(x) = \{1+\} - F_A(x), \]

for all $x$ in $X$.

- **Union**: The union of two neutrosophic sets $A$ and $B$ denoted by $A \cup B$ and defined as follows Smarandache (2003), Wang et al. (2005), Arora et al. (2011):

\[ T_{A \cup B}(x) = T_A(x) + T_B(x) - T_A(x) \times T_B(x), \quad I_{A \cup B}(x) = I_A(x) + I_B(x) - I_A(x) \times I_B(x), \quad F_{A \cup B}(x) = F_A(x) + F_B(x) - F_A(x) \times F_B(x). \]

for all $x$ in $X$.

- **Intersection**: The intersection of two neutrosophic sets $A$ and $B$ denoted by $A \cap B$ and defined as follows Smarandache (2003), Wang et al. (2005), Arora et al. (2011):

\[ T_{A \cap B}(x) = T_A(x) \times T_B(x), \quad I_{A \cap B}(x) = I_A(x) \times I_B(x), \quad F_{A \cap B}(x) = F_A(x) \times F_B(x). \]

for all $x$ in $X$.

- **Containment**: A neutrosophic set $A$ is contained in another neutrosophic set $B$ ($A \subseteq B$) if and only if (Smarandache, 2003; Wang et al., 2005)

\[ \inf f_{T_A} \leq \inf f_{T_B}; \quad \sup I_A \leq \sup I_B, \quad \sup F_A \leq \sup F_B. \]
\[ \inf_{T_x}(x) \leq \inf_{F_x}(x); \sup_{T_x}(x) \leq \sup_{F_x}(x). \]

where \( \sup_{T_x}, \sup_{I_x} \) and \( \sup_{F_x} \) represent the supremum for the \( T_x, I_x \) and \( F_x \), respectively, and \( \inf_{T_x}, \inf_{I_x} \) and \( \inf_{F_x} \) are the infimum of the \( T_x, I_x \) and \( F_x \), respectively, and \( x \in X \).

3.1.2. Neurotrophic logic

Neurotrophic logic was developed to represent the mathematical models that contains uncertainty, vagueness, ambiguity, imprecision, incompleteness, inconsistency, redundancy as well as contradiction (Hassanien, Basha, & Abdalla, 2018; Smarandache, 2003). Neurotrophic logic is a logic in which each proposition is estimated to have a percentage of truth in a subset \( T \), a percentage of indeterminacy in a subset \( I \), and a percentage of falsity in a subset \( F \), where \( T, I, F \) are standard or non-standard real subsets of \([0, 1]^I\) where \([0, 1]^I\) is a non-standard unit interval (Ansari et al., 2013; Robinson, 2003). \( T, I, F \) and \( x \) are called neurotrophic components, and these components represent the truth, indeterminacy, and falsehood values respectively in studying neurotrophic logic, neurotrophic set, neurotrophic probability, neurotrophic statistics (Ashbacher, 2002). In real world applications, it is easier to use standard real interval \([0,1] \) for \( T, I, F \) instead of the non-standard unit interval \([0, 1]^I\) (Ansari et al., 2013; Basha, Sahlol, El Baz, & Hassanien, 2017).

The sets \( T, I, F \) are not necessarily intervals, but may be any real sub-units subsets: discrete or continuous; single-element, finite, or (countable or uncountable) infinite; union or intersection of various subsets (Basha, Abdalla, & Hassanien, 2016b; Smarandache, 2003). Statically, \( T, I, F \) are subsets, but dynamically the components \( T, I, F \) and \( x \) are set-valued functions/operators depending on many parameters, such as: time, space, etc. (Smarandache, 2003).

3.2. Neurotrophic rule-based classification system

In 2013 Ansari et al. (2013) presented a neurotrophic classifier as an extension to fuzzy classifier using the Matlab software. The NRCS first appeared in 2016, by Basha et al. (2016b). An application of this system was presented in Basha et al. (2017) to predict the pollution status of the Burullus lagoon, in Egypt, according to the concentrations of trace metals. In 2018, another application introduced also by Basha, Tharwat, Ahmed, and Hassanien (2018), for building a predictive model to estimate the sperm quality based on personal lifestyle and environmental factors using NRCS. And in these applications, the NRCS always had better results compared to other models. This is because the NRCS model gives a good solution for the overlapped classes by generating three different components, and two of these components deal with the falsity and indeterminacy in the data.

The proposed Neurotrophic Rule-based Classification System (NRCS) uses Neurotrophic Logic (NL) for generalizing the fuzzy rule-based classification system. The antecedents and consequents of the “IF-THEN” rules in the NRCS are neurotrophic logic statements, instead of fuzzy logic ones. The NRCS has three stages:

1. **Neurotrophication:** construction of the neurotrophic knowledge-base by converting crisp inputs using the neurotrophic three membership functions: truth-membership, falsity-membership, and indeterminacy-membership.

2. **Inference Engine:** The KB and neurotrophic “IF-THEN” rules are applied to get a neurotrophic output.

3. **Deneurotrophication:** Converts the neurotrophic output of the previous step back to a crisp value using three functions analogous to the ones used by the neurotrophication.

The used knowledge base (KB) stores the available knowledge in the form of neurotrophic “IF-THEN” rules, and then the KB captures the neurotrophic rule semantics using neurotrophic sets. Fig. 1 shows the NRCS consisting of four phases: information extraction phase, neurotrophication phase, rules generation phase, and the classification phase. More details about each phase in the following subsections.

3.2.1. Information extraction phase

In this phase, important information are extracted by reading data files and extracting (1) the number of attributes, (2) the minimum and the maximum value of each attribute, (3) the number of classes and their names, and (4) the class labels or decisions.

3.2.2. Neurotrophication phase

In this phase, the three membership functions, namely, truth, falsity, and indeterminacy are extracted from the fuzzy-Trapezoidal membership function. Then, these three membership functions are applied on each value for every attribute in the dataset to obtain the neurotrophic components \( < T; I; F > \) that are used to represent each of every feature.

3.2.3. Rules generation phase

The goal of this phase is to generate rules which will be used in the next phase (classification phase). Assume the data is denoted by \( X = \{x_1, x_2, \ldots, x_n\} \), where \( x_i \) is the \( i \)th sample and \( n \) is the total number of samples. Each sample has one class label which is denoted by \( c_i \in \{1, 2, \ldots, C\} \), where \( C \) is the total number of classes. First, the dataset is divided into training data \( X_{\text{training}} \) and testing data \( X_{\text{testing}} \). In this phase, neurotrophic rules are generated from the training and testing data. The training rules are denoted by \( R_{\text{training}} = \{r_{1r}, r_{2r}, \ldots, r_{nr}\} \), where \( r_{ir} \) is the rule for the \( i \)th training sample and \( n_r \) is the number of training samples. Similarly, the testing rules are denoted by \( R_{\text{testing}} = \{r_{1t}, r_{2t}, \ldots, r_{nt}\} \), where \( r_{it} \) is the rule for the \( i \)th testing sample and \( n_t \) is the number of testing samples. In NRCS, the attribute in each neurotrophic rule has three components \( < T; I; F > \).

3.2.4. Classification phase

In this phase, for each testing rule \( r_{it} \in R_{\text{testing}} \), the Euclidean distance between a testing rule and all training rules \( R_{\text{training}} \) is calculated. As shown in Fig. 1, a vector of distance scores is calculated. The class label of the training rule which has the minimum distance is assigned to the testing rule.

Finally, we use the confusion matrix in Fig. 2 to evaluate the proposed model. From the confusion matrix, different measures can be calculated such as True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN).

Fig. 3 shows an example for comparing the fuzzy classifier and the neurotrophic classifier. There are two classes (class I and class II), each class has many samples. As shown in Fig. 3(a), there are two different types of outputs. The first type when the output clearly lies in one of the two classes. This is clear in Fig. 3(b), where the output of the fuzzy classifier is (1) 100% belongs to class I in the range between 0 and a, and (2) 100% belongs to class II in the range between b and c. The overlapping zone in gray color has the second type of outputs. In this region, as shown, there is a certain degree of indeterminacy, and this region has three possible outputs, (1) high membership values of class I in the range between \( a \) and \( a + b)/2 \), (2) high membership values of class II in the range between \( a + b)/2 \) and \( b \), and (3) equal membership values for the two classes at the point \( (a + b)/2 \). Moreover, in the overlapping region between class I and class II, the fuzzy membership function decreases till it reaches the point \( a + b)/2 \) where the two classes have the same membership function values. Fig. 3(c) shows the truth component of the neurotrophic class indicating 100% belongingness to class I in the range between 0 and \( a \), and 100%
belongingness to class II in the range between b and c, same result obtained by the fuzzy classifier. Additionally, in Fig. 3(c), the truth membership function/component has non-overlapping zone, and the overlapping region in the neutrosophic classifier is captured by the falsity and indeterminacy components as shown in Fig. 3(d) and (e), respectively.

Hence, the fuzzy classifier depends only on the membership of a particular sample to a particular class, and it does not deal with the indeterminate nature of the data. On the other hand, the neutrosophic classifier has two components which deal with the falsity and indeterminacy components to handle the overlapping region between any two classes.

### 3.2.5. Illustrative example

The goal of this example is to explain the steps of the NRCS model. Table 3 illustrates the data which is used in this example. As mentioned before, the first phase of the NRCS model is the information extraction phase. This phase extracts the following information:

- The number of attributes is two (f₁ and f₂).

We divided the data into two parts: training data (in black color in Table 3) and testing data (in red color). The training and the testing sets have the same number of samples (five samples).

In the second phase of the NRCS model (i.e., Neutrosophication phase), all values in the dataset are mapped to the neutrosophic space. This means that each value will be represented by three values/components (t, i, f) using the neutrosophic membership functions T, I, and F. Table 4 shows samples of the neutrosophic values of the example. The table shows only four values due to the length of the paper constraints. As shown, each value is converted as follows: \(< t_{\text{Low}}, t_{\text{Medium}}, t_{\text{High}} >, < i_{\text{Low}}, i_{\text{Medium}}, i_{\text{High}} >, < f_{\text{Low}}, f_{\text{Medium}}, f_{\text{High}} >\). For example, 0.2, the first value of the first attribute in
Table 4
Samples of the data of the illustrative example mapped to the neutrosophic space.

<table>
<thead>
<tr>
<th>Value</th>
<th>( t_{low} )</th>
<th>( t_{mid} )</th>
<th>( t_{high} )</th>
<th>( f_{low} )</th>
<th>( f_{mid} )</th>
<th>( f_{high} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
</tr>
<tr>
<td>5.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>2.5</td>
<td>0.0</td>
<td>0.667</td>
<td>0.0</td>
<td>0.334</td>
<td>0.0</td>
<td>1.0</td>
</tr>
<tr>
<td>6.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>3.2</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>3.0</td>
<td>0.667</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
</tbody>
</table>

In the rule generation phase (third phase), the training and testing rules are generated from the data in the neutrosophic space (see Table 4). The training rules are used later for classifying unseen/testing data. In this example, the generated training rules are as follows:

\[
[1.0, 0.0][3.0, 0.0] \rightarrow \text{class A} \\
[2.4, 7][3.0, 0.0] \rightarrow \text{class A} \\
[2.0, 0.0][1.0, 0.0] \rightarrow \text{class A} \\
[2.0, 0.0][2.0, 0.0] \rightarrow \text{class B} \\
[2.0, 0.0][3.0, 0.0] \rightarrow \text{class B} \\
\]

(1)

The first rule can be read as: if \( f_1 \) is [Low, 0, 0] and \( f_2 \) is [High, 0, 0] then the class label for this sample is A, and similar interpretations for the other rules. The testing rules are as follows:

\[
[1.0, 0.0][1.0, 0.0] \\
[1.0, 0.0][1.0, 0.0] \\
[3.0, 0.0][1.0, 0.0] \\
[3.0, 0.0][1.0, 0.0] \\
[3.0, 0.0][3.0, 0.0] \\
\]

(2)

In the last phase in the NRCS model, i.e., the classification phase, each rule in the testing data is matched with all the training rules (see Fig. 1). In this phase, we used the Euclidean distance for measuring the distance between the testing rule and the training rules. The testing rule belongs to the class which has the training rule that has the minimum distance to the testing rule. In other words, we assigned the class label of the training rule which has the minimum distance to the testing rule. For example, the distance between first rule in testing data (i.e., first testing rule) and the first rule in the training data (i.e., first training rule) is calculated as follows, \( \sqrt{(1-1)^2 + (0-0)^2 + (0-0)^2} + \sqrt{(1-3)^2 + (0-0)^2 + (0-0)^2} = 2 \). Similarly, the distances from this testing rule to all the training rules are calculated. The nearest training rule to this testing rule was the third training rule which means that this testing rule belongs to the class A. Hence, the class label of the first testing rule is A. In our example, three samples were correctly classified and the third and fourth testing samples were misclassified.

3.3. GNRCs: Hybrid classification system based on neutrosophic logic and genetic algorithm

The proposed Genetic Neutrosophic Rule-Based Classification System (GNRCS) is a hybridization of the NRCS model and the Genetic Algorithm (GA). In the GNRCs model, the GA is used for refining the neutrosophic “IF-THEN” rules as shown in Fig. 4. Hence, in this model, a new phase is added and it is called Genetic-based machine learning phase based on the Michigan approach (Elhoseny, Tharwat, & Hassanien, 2018; Metawa, Hassan, & Elhoseny, 2017). The GNRCs algorithm uses the same steps of the NRCS algorithm for generating the training and testing rules. In NRCS, the rules are generated automatically and may include redundant rules; here the GA is used for generating new rules and hence GA
explores the rules space to find the most effective rules and remove the redundant rules.

The GA parameters such as crossover probability ($P_c$), mutation probability ($P_m$), population size ($N$), number of iterations ($T$), and stopping condition(s) are first initialized. In the proposed model, the population size is the number of generated rules from the NRCS classifier. As shown in Fig. 4, the solutions are evaluated by calculating the misclassification rate which is the ratio between the number of misclassified samples ($N_e$) to the total number of testing samples ($N$), as follows:

$$\min J = \frac{N_e}{N} \quad (3)$$

The GA generates new rules if the termination criteria are not satisfied. In each round of the GA, the rules with misclassification rate higher than $50\%$ are removed; this is inspired from the Adaboost classifier (Schapire, Freund, Bartlett, Lee et al., 1998).

In the GA, when the termination criteria are satisfied, the operation ends; otherwise, we proceed with the next generation operation. In the proposed model, the GA is terminated when the best solution is not modified for a given number of iterations or when a maximum number of iterations are reached. In our experiments, the maximum number of repetitions of the best solution was five. Algorithm 1 summarizes the steps of the GNRCs model.

**Algorithm 1** GNRCs model.

1: Parameters initialization: number of rules ($N$), the crossover probability ($P_c$), the mutation probability ($P_m$), and the stopping condition.
2: Generates the initial population from the data using NRCS.
3: Evaluate the fitness value for all rules in the current population.
4: while stopping condition is not satisfied do
5: Using GA for finding new population through crossover and mutation processes.
6: Evaluate the fitness value for each rule in the current population.
7: Select the rules/solutions which obtain the minimum fitness values.
8: Remove the rules which have misclassification rate higher than $50\%$.
9: end while

3.4. Feature selection using rough set theory

The Rough set (RS) theory is one of the mathematical approaches that is used to deal with imprecision and uncertainty (Inbarani, Azar, & Jothi, 2014). Information System (IS) represents data as a table where each row represents one object and each column indicates one feature. Mathematically, IS can be defined as follows, $I = (U, A, V, f)$, where $U$ is a non-empty finite set of objects, $A$ represents a non-empty finite set of features, $V = \cup_{a \in A} V_a$ represents the union of the features domain, and a function $f_a: U \rightarrow V_a$ where $V_a$ is the set of values of feature $a$. The Decision System (DS) has the same structure as the IS, but each object has its own decision. For example, in our toxicity dataset, each object/row has a set of features and a decision of that object, whether this object is toxic or not. Hence, the decision system $S = (U, A, \cup d, V, f)$, where $A$ is the condition features and $d$ represents a decision feature (Chen, Miao, & Wang, 2010; Shen, Zhu, & Xu, 2015; Pawlak, 1982; Wang, Yang, Teng, Xia, & Jensen, 2007).

Each non-empty subset $P \subseteq A$ determines an equivalence relation as follows:

$$IND(P) = \{(x, y) \in U \times U | \forall a \in P, f_a(x) = f_a(y)\} \quad (4)$$

If $(x, y) \in IND(P)$, then $x$ and $y$ are indiscernible by attributes from $P$. The partition of $U$ generated by $P$ is as follows, $U/P = \{[x]_P | x \in U\}$, where $[x]_P$ indicates the equivalent class of the $P$-indiscernibility relation (Chen et al., 2015; Wang et al., 2007). The lower and the upper approximations of the set $X \subseteq U$ are denoted by $\overline{P_X} = \{x \in U | [x]_P \subseteq X\}$ and $\overline{P_X} = \{x \in U | [x]_P \cap X \neq \emptyset\}$, respectively. Let $P_1, P_2 \subseteq A$ be equivalent relations over $U$, the positive, negative, and boundary regions are defined, respectively, as follows: $POS_{P_1}(P_2) = \overline{P_{X_{P_1}}} \cap \overline{P_{X_{P_2}}}$. $NEG_{P_1}(P_2) = U - \overline{P_{X_{P_1}}} \cap \overline{P_{X_{P_2}}}$. $BND_{P_1}(P_2) = \overline{P_{X_{P_1}}} \cap \overline{P_{X_{P_2}}}$. Each rule of the relation $U/P$ with respect to $P$, $NEG_{P_1}(P_2)$ represents the negative region, and $BND_{P_1}(P_2)$ represents the boundary region. The set is called rough if it has a non-empty boundary region (Chen et al., 2015; Wang et al., 2007).

Measuring dependencies between a set of features is one of the most important tasks of the data analysis. Given $P, Q \subseteq A$, and all features from the relation $P$ are determined by the features from $Q$ and hence if there is a relation between $P$ and $Q$ then $P$ depends totally on $Q (IND(P) \subseteq IND(Q))$, this dependency is denoted by $Q \rightarrow P$. The degree of dependency is denoted by $k$ and is calculated as follows, $k = \gamma(Q) = \frac{|POS_{P_1}(Q)|}{|U|}$, where $|U|$ is the cardinality of $U$ and

- If $k = 1$, then $P$ depends totally on $Q$;
- if $k = 0$, then $P$ does not depend on $Q$;
- if $0 < k < 1$, then $P$ depends partially on $Q$ (Chen et al., 2015; Wang et al., 2007).

The main goal of the feature reduction methods is to remove redundant features so that the reduced set achieves the same classification performance as the original features. The reduct is a minimal subset $R$ of the original features $C$ such that $\gamma_C(R) = \gamma_C(D)$, where $R$ is the minimal subset if $\gamma_C(R) \neq \gamma_C(D)$, $\forall a \in R$. This means that there is no features could be removed from $R$ without affecting the dependency degree. Rough sets are used in finding the reduct with the smallest cardinality that represents the global minimum $\{R_{min} = \{R \in R_{all}, \forall Y \in R_{all}, |R| \leq |Y|\}\}$ (Chen et al., 2015; Wang et al., 2007).

In this paper, three different rough set-based methods are utilized for feature selection, namely, Quick Reduce Feature Selection (QFRS) (Jensen & Shen, 2003), Discriminability Matrix-based Feature Selection (DMFS) (Wang, Miao, & Hu, 2006), and Entropy-based Selection (EBFS) (Jensen & Shen, 2003). We have selected these methods because each method has different strategy to find the minimal reduct. The DMFS method, is one of the original rough set-based methods and it was introduced by Pawlak (1991). The EBFS method is based on the entropy heuristic employed by a machine learning algorithm such as C4.5. The QFRS method tries to find the minimal reduct without exhaustively generating all possible subsets. Due to paper length restrictions, we will not describe
these algorithms here; more details can be found in the related references. However, these algorithm obtained competitive results.

3.5. Imbalanced datasets

The problem of imbalanced datasets appears when the number of samples of one class (majority class ($S_{maj}$)) is significantly higher than the samples of the other class (minority class ($S_{min}$)) (He & Garcia, 2009; Sun, Wong, & Kamel, 2009). This problem is found frequently in the classification problems and it decreases the classification performance, and this is due to many reasons, (1) it is difficult for a learning algorithm to learn from a minority class; hence, the minority class samples are misclassified frequently; (2) using global assessment methods to evaluate learning algorithms may provide an advantage to the majority class (López, Fernández, García, Palade, & Herrera, 2013).

There are many methods used to handle the imbalanced data problem such as sampling methods (He & Garcia, 2009), Kernel-Based methods (He & García, 2009), and Cost-Sensitive methods (Sun et al., 2009). In this paper, the sampling methods are used to obtain more balanced samples in each class.

The cost-sensitive methods are not easy to implement and there are some learning algorithms such as C4.5 do not directly handle costs in the learning process (Weiss, McCarthy, & Zabar, 2007). Moreover, the performance of the cost-sensitive methods is highly affected by the evaluation metrics (Weiss et al., 2007). The kernel functions are used in many machine learning techniques. For example, using SVM, the kernel functions are used to transform the nonlinearly separable data into a higher dimensional space where the data can be linearly separable. With the imbalanced data, the SVM learning model classifies the majority class samples better than the minority class samples. In this case, the kernel can be used and the optimal hyperplane will be biased towards the majority class. However, the sampling methods are simpler than the cost-sensitive and the kernel-based methods.

In the sampling methods, the goal is to modify the prior distribution of the majority and the minority classes to obtain more balanced samples in each class. There are three well-known sampling methods: Random Under-sampling (RUS), Random Over-sampling (ROS), and Synthetic Minority Over-sampling Technique (SMOTE). In the RUS method, the goal is to extract a small set of the majority class samples for training a classifier while preserving all the minority class samples. As a consequence, the training data becomes more balanced and smaller and hence the required computational time is less. However, the removed samples may have useful information and hence RUS may decrease the classification performance (He & Garcia, 2009). In the ROS method, the goal is to replicate the minority class samples. Hence, this method improves the minority class recognition. However, making exact copies of minority class samples increases the learning time and may lead to the overfitting problem (He & Garcia, 2009; Sun et al., 2009). In the SMOTE method, the minority samples are created based on the similarities between existing minority samples. For each minority sample ($x_i \in S_{min}$), k nearest samples are selected and a synthetic sample can be generated as follows, $x_{new} = x_i + \delta \times 1 = x_i + (x_{ij} - x_i) \times \delta$, where $x_i \in S_{min}$ is one of the minority class samples, $x_{ij}$ is one of the k-nearest samples for $x_i : x_{ij} \in S_{min}$, $j = 1, 2, \ldots, k$. k is the number of selected neighbors, $\delta \in [0, 1]$ is a random number, and $x_{new}$ is a sample along the line joining $x_i$ and $x_{ij}$ (Chawla, Bowyer, Hall, & Kegelmeyer, 2002; He & Garcia, 2009).

4. Experimental results and discussions

In this section, four experiments were conducted. The first experiment (in Section 4.1) has four goals (1) Testing the proposed NRCS model for predicting the toxicity of the biotransformed drugs, (2) Testing the power of this system to deal with uncertain data without using any feature selection or any pre-processing method, (3) Comparing NRCS with conventional classifiers such as Multi-Layer Perceptron (MLP) (Yamany et al., 2015), k-Nearest Neighbors (k-NN) (Tharwat, Mahdi, Elhoseny, & Hassanien, 2018), and Linear Discriminant Analysis (LDA) (Tharwat, 2016) classifiers, and finally, (4) Comparing the proposed models (NRCS and GNRCs).

Different runs were conducted for finding the optimal or near optimal parameters for k-NN and MLP classifiers. Based on these runs we found that the value of k in k-NN was five and in MLP, the hidden layer with 15 nodes and 1000 epochs were used. The optimal values of the NRCS parameter were found to be: $N = 200$, $P_r = 0.9$, and $P_m = 0.1$ for the number of iterations, the crossover probability, and the mutation probability, respectively.

The aim of the second experiment (in Section 4.2) is to obtain balanced data for improving the sensitivity of the proposed model. In this experiment, three sampling methods were used, namely, RUS, ROS, and SMOTE algorithms. The aim of the third experiment in Section 4.3 is to reduce the number of features and hence reduce the required computational time by using three different rough set-based feature selection algorithms. The goal of the fourth experiment is to compare the NRCS and GNRCs algorithms using standard datasets. Moreover, in this experiment, we compared the performance of the NRCS and GNRCs algorithms with conventional classifiers such as MLP, k-NN, and LDA classifiers. In this experiment, the optimal parameters of the k-NN and MLP classifiers were obtained during the training phase.

In all experiments, the results were obtained with a $5 \times 10$-fold cross-validation. The samples of the dataset were randomly divided into $k = 10$ subsets of equal size, and the experiment was run 10 times. For each run, one subset was used for testing the model, and the other subsets were used for training the model. This process was repeated five times. The results are the average of these $5 \times 10$ experiments.

All experiments are performed using the same PC with the properties settings in Table 5. Moreover, all the algorithms in this research are self-coded using java.

4.1. Toxicity classification using the NRCS and GNRCs models

The aim of this experiment is to test the two proposed models NRCS and GNRCs for predicting the toxicity of biotransformed drugs. In this experiment, all attributes were used, i.e., without any feature selection method. In other words, in this experiment, the original data was used with no preprocessing steps. This experiment has two sub-experiments.

4.1.1. NRCS Vs. conventional classifiers

The aim of this comparison is to test the NRCS system against the imprecision, incompleteness, vagueness, and inconsistency in data. This experiment has two sub-experiments. In the first sub-experiment, we compared the results of NRCS with three well-known learning algorithms, MLP (Yamany et al., 2015), k-Nearest
### Table 6
A comparison between the results (accuracy, sensitivity, specificity, F1-Score, and GM) of the proposed model using NRCS and the results obtained from MLP, LDA, and k-NN classifiers.

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Mutagenic effect</th>
<th>Tumorigenic effect</th>
<th>Irritant effect</th>
<th>Reproductive effect</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NN</td>
<td>LDA</td>
<td>KNN</td>
<td>NRCS</td>
</tr>
<tr>
<td>Accuracy</td>
<td>81.82</td>
<td>83.64</td>
<td>76.36</td>
<td>87.27</td>
</tr>
<tr>
<td>Sensitivity</td>
<td>28.57</td>
<td>33.33</td>
<td>20.00</td>
<td>50.00</td>
</tr>
<tr>
<td>Specificity</td>
<td>89.58</td>
<td>89.80</td>
<td>88.89</td>
<td>91.48</td>
</tr>
<tr>
<td>F1-Score</td>
<td>28.57</td>
<td>30.77</td>
<td>23.53</td>
<td>46.15</td>
</tr>
<tr>
<td>GM</td>
<td>50.59</td>
<td>54.71</td>
<td>42.16</td>
<td>67.76</td>
</tr>
</tbody>
</table>

### Table 7
A comparison between the results (accuracy, sensitivity, specificity, F1-Score, and GM) of the proposed model using NRCS and the results obtained from Bagging (Bag), AdaBoost (Ada), and Random forest (RF) ensemble.

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Mutagenic effect</th>
<th>Tumorigenic effect</th>
<th>Irritant effect</th>
<th>Reproductive effect</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Bag</td>
<td>Ada</td>
<td>RF</td>
<td>NRCS</td>
</tr>
<tr>
<td>Accuracy</td>
<td>84.50</td>
<td>84.00</td>
<td>86.64</td>
<td>87.27</td>
</tr>
<tr>
<td>Sensitivity</td>
<td>38.57</td>
<td>40.27</td>
<td>46.87</td>
<td>50.00</td>
</tr>
<tr>
<td>Specificity</td>
<td>87.54</td>
<td>88.47</td>
<td>90.25</td>
<td>91.48</td>
</tr>
<tr>
<td>F1-Score</td>
<td>36.78</td>
<td>40.54</td>
<td>42.98</td>
<td>46.15</td>
</tr>
<tr>
<td>GM</td>
<td>57.89</td>
<td>60.57</td>
<td>64.67</td>
<td>67.76</td>
</tr>
</tbody>
</table>

Neighbors (k-NN) (Tharwat et al., 2018), and Linear Discriminant Analysis (LDA) (Tharwat, 2016).

The ensemble methods, which have received much attention in recent years, have been known to be accurate and robust to any noise in data. They introduced a nice idea for combing weak machine learning algorithms to produce a stronger one. The learned model of this ensemble is a collection of the underlying models capturing the best of them. This new model lacks explanations and hence transparency of its results (Al Iqbal, 2012; Friedman, Popescu et al., 2008). Examples of these ensembles are the bagged ensemble (Breiman, 1996), the Random forests (Breiman, 2001), and the AdaBoost (Freund, Schapire et al., 1996). Hence, in the second sub-experiment, we applied the ensemble methods such as Bagging, Random Forest, and AdaBoost, and we compared the results of the NRCS with their results.

The results of these experiment in terms of accuracy, sensitivity, specificity, F1-score, and GM⁴ are summarized in Tables 6 and 7. From Table 6 the following remarks can be drawn.

- In the mutagenic effect, the NRCS obtained the best results, and the LDA classifier obtained the second-best results. While the k-NN classifier achieved the worst results.
- In the Tumorigenic effect, the NRCS classifier obtained the best results and the LDA classifier obtained competitive results. As shown in Table 6, the NRCS and LDA classifiers obtained the same specificity; however, the NRCS classifier obtained better sensitivity score than LDA. While the k-NN classifier obtained also the worst results.
- In the irritant effect, the NRCS algorithm achieved the best results and the MLP and LDA classifiers obtained competitive results. The specificity of the MLP and NRCS classifiers was better than the other two classifiers.
- In the reproductive effect, the NRCS and MLP classifiers obtained the best accuracy and GM results. However, MLP has achieved sensitivity results better than NRCS, while NRCS obtained specificity and F1-Score better than MLP.

The results of the second sub-experiment in Table 7 show that the proposed NRCS obtains overall the best results. In the mutagenic effect, the NRCS obtains the best results and the random forest classifier obtained the second best results. In the Tumorigenic effect, also the NRCS achieves the best results and the random forest classifier obtains the second best results. In the irritant effect, the bagging classifier obtains the best results and the NRCS obtains the second best results while the random forest obtains competitive results. Finally, in the reproductive effect, the NRCS obtains the best results.

Therefore, the proposed NRCS achieves overall the best results and the LDA and MLP classifiers have competitive results in the first sub-experiment, and the ensemble methods obtain also competitive results. And it is a natural result - after using the inde terminacy term in the neutrosophic logic - to have the NRCS determines the more significant, neutral and non-significant features without any feature selection method. Also, it is clear that the sensitivity results were much lower than the specificity results. This is because the data was imbalanced; and from Table 2, the reproductive effect has the minimum imbalance ratio. Which is also the reason why the sensitivity results of the reproductive effect were much higher than the other toxicity effects.

#### 4.1.2. NRCS vs. GNCRS

The aim of using the hybrid classification system (GNCRS) is to improve the results obtained from the first sub-experiment for predicting the toxicity of the biotransformed drugs. As in the first sub-experiment, all attributes were used, i.e., without any feature selection method. The results of this experiment were compared with the result of the first sub-experiment in terms of accuracy, sensitivity, specificity, F1-Score, and GM and were summarized in Table 8. It is worth mentioning that the results of the GNCRS model represent the best results obtained.

From Table 8 it can be noted that in terms of accuracy, sensitivity, F1-Score and GM metrics, the proposed GNCRS algorithm outperformed the NRCS algorithm in most cases. Moreover, in terms of specificity, the GNCRS and NRCS algorithms obtained competitive results. In other words, the GNCRS algorithm improved the sensitivity more than the specificity. This is because GA in the GNCRS algorithm searches for the optimal or near optimal rules and removes redundant rules; hence, the GA refines the generated neurotropic rules. Additionally, in this experiment, we obtained high prediction accuracy with less number of training rules. Table 9 shows the number of training rules for both NRCS and GNCRS models.

---

⁴ More details about these classification metrics are in (Tharwat, 2018).
Table 8

A comparison between the results (accuracy, precision, sensitivity, specificity, F1-Score, and GM) of the proposed model using GNRCs and the results obtained from NRCS.

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Mutagenic effect</th>
<th>Tumorigenic effect</th>
<th>Irritant effect</th>
<th>Reproductive effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>NRCS</td>
<td>GNRCs</td>
<td>NRCS</td>
<td>GNRCs</td>
<td>NRCS</td>
</tr>
<tr>
<td>Accuracy</td>
<td>87.27</td>
<td>89.09</td>
<td>85.46</td>
<td>87.27</td>
</tr>
<tr>
<td>Sensitivity</td>
<td>50.00</td>
<td>57.14</td>
<td>44.44</td>
<td>48.00</td>
</tr>
<tr>
<td>Specificity</td>
<td>91.48</td>
<td>93.75</td>
<td>93.48</td>
<td>93.62</td>
</tr>
<tr>
<td>F1-Score</td>
<td>46.15</td>
<td>57.14</td>
<td>50.00</td>
<td>53.33</td>
</tr>
<tr>
<td>GM</td>
<td>67.76</td>
<td>73.19</td>
<td>64.46</td>
<td>68.42</td>
</tr>
</tbody>
</table>

Table 9

Number of training rules of the proposed models (NRCS and GNRCs) with all datasets.

<table>
<thead>
<tr>
<th># NRCS rules</th>
<th># GNRCs rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>304</td>
<td>282</td>
</tr>
<tr>
<td>277</td>
<td>229</td>
</tr>
</tbody>
</table>

Table 10

A comparison between the proposed algorithms (NRCS and GNRCs) and two from the state-of-the-art approaches (DA+SVM [Tharwat, Gabel et al., 2017] and WOA+SVM [Tharwat, Moemen et al., 2017]) with different sampling algorithms in terms of sensitivity, specificity, and GM metrics using mutagenic effect.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Metrics</th>
<th>Orig.</th>
<th>RUS</th>
<th>ROS</th>
<th>SMOTE</th>
</tr>
</thead>
<tbody>
<tr>
<td>DA+SVM (Tharwat, Gabel et al., 2017)</td>
<td>Sensitivity</td>
<td>48.00</td>
<td>54.19</td>
<td>86.49</td>
<td>89.43</td>
</tr>
<tr>
<td></td>
<td>Specificity</td>
<td>86.60</td>
<td>75.43</td>
<td>82.46</td>
<td>86.55</td>
</tr>
<tr>
<td></td>
<td>GM</td>
<td>64.47</td>
<td>63.93</td>
<td>84.45</td>
<td>88.09</td>
</tr>
<tr>
<td>WOA+SVM (Tharwat, Moemen et al., 2017)</td>
<td>Sensitivity</td>
<td>50.00</td>
<td>57.23</td>
<td>85.43</td>
<td>89.74</td>
</tr>
<tr>
<td></td>
<td>Specificity</td>
<td>88.50</td>
<td>76.84</td>
<td>83.24</td>
<td>87.27</td>
</tr>
<tr>
<td></td>
<td>GM</td>
<td>66.52</td>
<td>66.31</td>
<td>84.33</td>
<td>88.50</td>
</tr>
<tr>
<td>NRCS</td>
<td>Sensitivity</td>
<td>50.00</td>
<td>55.56</td>
<td>88.64</td>
<td>91.11</td>
</tr>
<tr>
<td></td>
<td>Specificity</td>
<td>91.48</td>
<td>77.78</td>
<td>85.71</td>
<td>89.58</td>
</tr>
<tr>
<td></td>
<td>GM</td>
<td>67.76</td>
<td>65.73</td>
<td>87.16</td>
<td>90.34</td>
</tr>
<tr>
<td>GNRC</td>
<td>Sensitivity</td>
<td>57.14</td>
<td>62.50</td>
<td>86.96</td>
<td>91.36</td>
</tr>
<tr>
<td></td>
<td>Specificity</td>
<td>93.75</td>
<td>80.00</td>
<td>87.23</td>
<td>91.49</td>
</tr>
<tr>
<td></td>
<td>GM</td>
<td>73.19</td>
<td>70.70</td>
<td>87.10</td>
<td>91.40</td>
</tr>
</tbody>
</table>

Table 11

A comparison between the proposed algorithms (NRCS and GNRCs) and two from the state-of-the-art approaches (DA+SVM [Tharwat, Gabel et al., 2017] and WOA+SVM [Tharwat, Moemen et al., 2017]) with different sampling algorithms in terms of sensitivity, specificity, and GM metrics using tumorigenic effect.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Metrics</th>
<th>Orig.</th>
<th>RUS</th>
<th>ROS</th>
<th>SMOTE</th>
</tr>
</thead>
<tbody>
<tr>
<td>DA+SVM (Tharwat, Gabel et al., 2017)</td>
<td>Sensitivity</td>
<td>46.42</td>
<td>60.74</td>
<td>90.84</td>
<td>90.05</td>
</tr>
<tr>
<td></td>
<td>Specificity</td>
<td>91.62</td>
<td>67.67</td>
<td>85.66</td>
<td>86.57</td>
</tr>
<tr>
<td></td>
<td>GM</td>
<td>62.33</td>
<td>64.11</td>
<td>88.21</td>
<td>88.29</td>
</tr>
<tr>
<td>WOA+SVM (Tharwat, Moemen et al., 2017)</td>
<td>Sensitivity</td>
<td>46.24</td>
<td>60.49</td>
<td>89.28</td>
<td>90.49</td>
</tr>
<tr>
<td></td>
<td>Specificity</td>
<td>94.65</td>
<td>68.84</td>
<td>84.54</td>
<td>87.64</td>
</tr>
<tr>
<td></td>
<td>GM</td>
<td>66.16</td>
<td>64.53</td>
<td>86.88</td>
<td>89.05</td>
</tr>
<tr>
<td>NRCS</td>
<td>Sensitivity</td>
<td>44.44</td>
<td>62.50</td>
<td>91.11</td>
<td>91.30</td>
</tr>
<tr>
<td></td>
<td>Specificity</td>
<td>93.48</td>
<td>70.00</td>
<td>87.50</td>
<td>89.36</td>
</tr>
<tr>
<td></td>
<td>GM</td>
<td>64.46</td>
<td>66.14</td>
<td>89.29</td>
<td>90.33</td>
</tr>
<tr>
<td>GNRC</td>
<td>Sensitivity</td>
<td>50.00</td>
<td>66.67</td>
<td>93.33</td>
<td>91.49</td>
</tr>
<tr>
<td></td>
<td>Specificity</td>
<td>93.62</td>
<td>77.78</td>
<td>89.58</td>
<td>91.30</td>
</tr>
<tr>
<td></td>
<td>GM</td>
<td>68.42</td>
<td>72.01</td>
<td>91.44</td>
<td>91.40</td>
</tr>
</tbody>
</table>

4.2. Obtaining balanced data

In this experiment, the goal is to obtain balanced data in both classes. To do this, three sampling algorithms were used: RUS, ROS, and SMOTE. In this experiment, (1) in the ROS algorithm, minority class samples were randomly oversampled until the number of minority class samples matched the number of majority class samples, (2) in the RUS algorithm, the majority class samples were randomly undersampled until their number matched the number of minority class samples, (3) in the SMOTE algorithm, the number of synthetic samples is a parameter in the SMOTE algorithm. In this experiment, samples of minority class were synthesized to equalize the two classes. The results of this experiment are summarized in Tables 10–13. From these tables many findings can be summarized as follows:

- In terms of sensitivity, all sampling methods obtained high sensitivity than the original data (Orig.). This is because, in our experiments, the positive class is the minority class. Hence, the sensitivity results of the original data were much lower than the specificity results (see the results of the first experiment). However, all sampling algorithms obtained balanced data and hence improves the sensitivity results. Moreover, the SMOTE algorithm achieved the best sensitivity results. This is because (1) the RUS algorithm randomly removes majority class samples; hence, important or discriminative data may have been removed, (2) ROS replicates minority samples in the
same positions of the original samples and hence the minority class cannot extend its decision boundary. On the other hand, the SMOTE algorithm generates samples in different positions around the minority samples; thus, the SMOTE algorithm extends the decision boundary of the positive class into the negative class which improves the sensitivity of the proposed model. Tables 10–13 show that the sensitivity of the RUS algorithm is significantly better than the original data; on the other hand, in Table 12, the RUS algorithm obtained sensitivity approximately equal to the original data. This is because (1) the imbalance ratio of the reproductive data was not high (see Table 2) and hence the number of removed samples was much smaller than the other cases, i.e., mutagenic, tumorigenic, and irritant, (2) the number of minority samples in the irritant data is 67 samples; this means that 486 – 67 = 419 samples were removed which has a negative influence on the overall results as shown in Table 13. In other words, removing a large number of samples makes the dataset small which is not sufficient for training the model.

- In terms of specificity, the results of RUS were lower than the original data. This is due to the samples that were removed from the majority class. ROS also obtained results lower than the original data in some cases. This is because in ROS, new minority samples were created and hence the two classes became balanced. As a result, the sensitivity increased as we mentioned before while the specificity has decreased. In most cases, the SMOTE algorithm obtained the best specificity results than the original data or the other two sampling algorithms. Further, the sensitivity and specificity goals are often conflicting; this is the reason why the results of the sampling methods improved the sensitivity but reduced the specificity. It is worth noticing that the SMOTE algorithm improved the sensitivity with a small reduction in the specificity results.

- In terms of GM, the results depend on both sensitivity and specificity results. As a consequence, all sampling algorithms obtained GM results better than the original data. Moreover, the SMOTE algorithm obtained the best results and the RUS algorithm has recorded the worst GM results.

Tables 10–13 show also a comparison between our proposed models and two from the state-of-the-art approaches (WOA+SVM (Tharwat, Moemen et al., 2017) and DA+SVM (Tharwat, Gabel et al., 2017)). It is interesting to note from the tables that the GNRCs model obtained the best results in most cases. This is because the two state-of-the-art approaches were based on stochastic nature. Therefore, there is no guarantee for finding the optimal solution. Moreover, the indeterminacy term in our proposed models makes the neutrosophic-based models able to determine the more significant, neutral and non-significant features without using any feature selection method. This is the reason why the GNRCs and NRCS models obtained better results than the other two models.

For more evaluation, a comparison between the sampling algorithms (RUS, ROS, and SMOTE) in terms of (between and within) class variances was conducted. The within-class variance is the difference between the mean and the samples of each class and it is defined as follows, $S_{ij} = \sum_{j=1}^{n_i}(x_{ij} - \mu_i)(x_{ij} - \mu_i)^T$, where $x_{ij}$ is the $j$th sample in the $i$th class, $S_{ij}$ is the within-class variance of the $i$th class, $n_i$ is the number of samples in the class $i$, and $\mu_i$ is the mean of the $i$th class. The total within-class variance is defined as follows, $S_W = \sum_{i=1}^{C}S_{ij}$, where $C$ is the number of classes.
The between-class variance is the distance between two or more classes, and it can be calculated by calculating the distance between the mean of each class \( (\mu_i) \) and the total mean of all classes \( (\mu) \). Our experiment shows that the SMOTE algorithm reduced the within-class variance and increased the between-class variance more than the other two sampling methods or the original data. This is because SMOTE generates samples around the minority class and hence SMOTE (1) reduces the distance between the samples of the minority class, i.e., reduces the within-class variance, (2) increases the distance between the generated samples and the samples of the majority class, i.e., increases the between-class variance. However, due to the randomness in the SMOTE algorithm, we still cannot guarantee that it must perform well with different datasets. Hence, the results of the SMOTE algorithm are affected if the original data is noisy, has many outliers, or the discrimination between classes is small (the classes are overlapped).

### 4.3. Feature selection using rough set

The aim of this experiment is to improve the results obtained from the first and the second experiments by using rough set-based methods. In this experiment, three different rough set-based methods, namely, QRFS, DMFS, and EBFS, were applied. Moreover, the three feature selection algorithms were tested with the NRCS and GNRCS algorithms. To obtain balanced data, we used the SMOTE algorithm which has achieved the best results in the second experiment. The results of this experiment are summarized in Tables 14–17.

Fig. 5 shows the selected features using all feature selection methods. As shown, many features were removed, and there is a big intersection between all feature selection algorithms. This intersection represents the most important features which have useful discrimination information. As shown, the twelfth and nineteenth features were selected by the three feature selection methods in all cases, which reflect the importance of these two features. Fig. 6 shows the scatter plot for each dataset using only the 12th and 19th features. Also, it is clear that the two classes are imbalanced because we used in this figure the original data. This resulted in low sensitivity and high specificity because in all experiments the positive class is assigned to the minority class.

From Tables 14–17, many remarks can be noticed. First, GNRCS obtained better results than NRCS in most cases, and this agrees with the conclusion of the first experiment. Second, all feature selection methods obtained better results than the original data, which also reduces the required computational time. Third, in most cases, the QRFS algorithm achieved better results than the other two feature selection algorithms.

In terms of computational time, the EBFS and QRFS algorithms required computational time lower than the DMFS algorithm. This is because the complexity of DMFS method is \( O((N + \log M)/M^2) \), where \( N \) and \( M \) indicate the number of features and samples, respectively. Thus, the DMFS algorithm needs a significant amount of time for the computation of the discernibility matrix, and the time was increasing fast by increasing the number of samples of the dataset. The complexity of EBFS and QRFS is \( O((N^2 + N)/2) \). In this paper, the number of features is \( N = 31 \) and the dataset has \( M = 553 \) samples; thus, QRFS and EBFS need computational time lower than DMFS.

### 4.4. NRCS vs. GNRCS using standard datasets

The goal of this experiment is to evaluate the performance of NRCS and GNRCS using different standard datasets. In this experiment, we compared the NRCS and GNRCS algorithms with three conventional classifiers such as MLP, \( k \)-NN, and LDA classifiers. We used three widely used standard classification datasets obtained from the University of California at Irvin (UCI) Machine Learning

---

**Table 14**

A comparison between the QRFS, DMFS, and EBFS features selection algorithms with the two proposed algorithms (NRCS and GNRCS) in terms of sensitivity, specificity, and GM metrics using mutagenic effect.

<table>
<thead>
<tr>
<th>Metrics</th>
<th>NRCS</th>
<th>GNRCS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity</td>
<td>Orig.</td>
<td>DMFS</td>
</tr>
<tr>
<td>Specificity</td>
<td>91.11</td>
<td>91.30</td>
</tr>
<tr>
<td>GM</td>
<td>90.34</td>
<td>91.40</td>
</tr>
</tbody>
</table>

**Table 15**

A comparison between the QRFS, DMFS, and EBFS features selection algorithms with the two proposed algorithms (NRCS and GNRCS) in terms of sensitivity, specificity, and GM metrics using tumorigenic effect.

<table>
<thead>
<tr>
<th>Metrics</th>
<th>NRCS</th>
<th>GNRCS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity</td>
<td>Orig.</td>
<td>DMFS</td>
</tr>
<tr>
<td>Specificity</td>
<td>91.30</td>
<td>93.33</td>
</tr>
<tr>
<td>GM</td>
<td>89.36</td>
<td>91.67</td>
</tr>
</tbody>
</table>

**Table 16**

A comparison between the QRFS, DMFS, and EBFS features selection algorithms with the two proposed algorithms (NRCS and GNRCS) in terms of sensitivity, specificity, and GM metrics using reproductive effect.

<table>
<thead>
<tr>
<th>Metrics</th>
<th>NRCS</th>
<th>GNRCS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity</td>
<td>Orig.</td>
<td>DMFS</td>
</tr>
<tr>
<td>Specificity</td>
<td>84.21</td>
<td>86.62</td>
</tr>
<tr>
<td>GM</td>
<td>88.89</td>
<td>91.43</td>
</tr>
</tbody>
</table>

---
Table 17
A comparison between the QRFS, DMFS, and EBFS features selection algorithms with the two proposed algorithms (NRCS and GNRCs) in terms of sensitivity, specificity, and GM metrics using irritant effect.

<table>
<thead>
<tr>
<th>Metrics</th>
<th>NRCS</th>
<th></th>
<th></th>
<th></th>
<th>GNRCs</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Orig.</td>
<td>DMFS</td>
<td>EBFS</td>
<td>QRFS</td>
<td>Orig.</td>
<td>DMFS</td>
<td>EBFS</td>
<td>QRFS</td>
</tr>
<tr>
<td>Sensitivity</td>
<td>83.67</td>
<td>85.71</td>
<td>89.36</td>
<td>89.80</td>
<td>89.36</td>
<td>89.80</td>
<td>91.84</td>
<td></td>
</tr>
<tr>
<td>Specificity</td>
<td>85.42</td>
<td>87.50</td>
<td>88.00</td>
<td>91.67</td>
<td>87.76</td>
<td>88.00</td>
<td>91.49</td>
<td></td>
</tr>
<tr>
<td>GM</td>
<td>84.54</td>
<td>86.60</td>
<td>88.68</td>
<td>90.73</td>
<td>88.56</td>
<td>88.68</td>
<td>90.64</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 5. Selected features of the three feature selection methods. (a) Mutagenic, (b) Tumorigenic, (c) Reproductive, (d) Irritant.

Fig. 6. Scatter plot for all datasets (a) Tumorigenic, (b) Irritant, (c) Reproductive, (d) Mutagenic. The scatter plot demonstrates the values of two of the most important selected features (12th and 19th features) which are highlighted in Fig. 5.

Repository (Blake, 1998). The descriptions of all datasets are as follows:
- The iris dataset has four features, three classes, and 150 samples,
- The wine dataset has 13 features, three classes, and 178 samples, and
- The Breast Cancer Wisconsin (Diagnostic) dataset (WDBC) has 32 features, two classes, and 569 samples.

The results of this experiment are listed in Table 18. Table 18 shows that the GNRCs algorithm obtained the best results, and the NRCS and LDA algorithms achieved competitive results. Moreover, the k-NN classifier achieved the worst results, and these results are in agreement with the results of the first experiment.

To conclude, the results of this paper indicate that the proposed algorithm (GNRCs) obtained better results than some conventional classifiers. This is because the NRCS model gives a good solution for the overlapped classes by generating three different components, and two of these components deal with the falsity and indeterminacy in the data. Moreover, in the proposed algorithm, the SMOTE algorithm obtained balanced data and hence improved the sensitivity of our model without sacrificing the specificity. Finally, the most discriminative features are selected us-
ing the QRFS algorithm which improved the classification performance and reduced the computational time. Overall, the proposed GNRCs with SMOTE and the QRFS feature selection algorithm obtained competitive results: sensitivity (89–93%), specificity (91.0–97.0%), and GM (90–94%). These results are better than the results obtained in (Tharwat, Moemen et al., 2017) [sensitivity (93.5 ± 8%), specificity (91.5 ± 7.7%), and GM (91.5 ± 7.7%)] and in (Tharwat, Gabel et al., 2017) [sensitivity (90.3 ± 7.6%), specificity (92.0 ± 8.3%), and GM (90.3 ± 6.8%)]. From the results, it is clear that due to the stochastic nature of the Whale algorithm and the Dragonfly algorithm in (Tharwat, Gabel et al., 2017; Tharwat, Moemen et al., 2017), the variation, i.e., standard deviation, of the results are high; on the other hand, our proposed model has no parameters that need to be tuned which makes our model more stable as mentioned in Section 1.

5. Conclusions and future work

This paper proposes a novel model for predicting drug toxicity risks. We used a dataset that has 553 drugs that biotransformed in the liver, and the data has four toxic effects, namely, mutagenic, tumorigenic, irritant, and reproductive. The proposed model has three main phases. First, in the feature selection phase, three rough set-based algorithms (Quick Reduct Feature Selection (QRFS), Discriminability Matrix-based Feature Selection (DMFS), and Entropy-based Selection (EBFS)) were used for selecting the most discriminative features. This step is important for removing the redundant features and hence reduces the computational efforts. Second, three data sampling algorithms, namely, Random Under-Sampling, Random Over-Sampling, and Synthetic Minority Over-sampling Technique (SMOTE) were used for obtaining balanced data. This step is important, as well, because the data that we used in our experiments was imbalanced. Third, in the classification phase, two novel classification algorithms were proposed, namely, Neutrosophic Rule-based Classification System (NRCS) and Genetic NRCS (GNRCS). Both models depend on generating neutrosophic rules and the goal is to classify an unknown drug into toxic or non-toxic. Different experiments were conducted for evaluating our model and the obtained results were promising. Moreover, the results proved that the proposed model obtained high sensitivity to all toxic effects. Overall, the results of the proposed model indicate that it could be utilized for the prediction of drug toxicity in the early stages of drug development.

Since the NRCS and the GNRCs classifiers have shown to be competitive with state-of-the-art classical classifiers, a promising future work would be applying the hybridization of the GA and the neutrosophic systems on ensemble-based classifiers.

### Table 18
A comparison between the NRCS, GNRCs, MLP, k-NN, and LDA classifiers using standard datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Metric</th>
<th>NRCS</th>
<th>GNRCs</th>
<th>MLP</th>
<th>k-NN</th>
<th>LDA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>Accuracy</td>
<td>96.40</td>
<td>99.30</td>
<td>95.40</td>
<td>90.20</td>
<td>96.50</td>
</tr>
<tr>
<td></td>
<td>Sensitivity</td>
<td>100.00</td>
<td>100.00</td>
<td>99.50</td>
<td>95.53</td>
<td>91.04</td>
</tr>
<tr>
<td></td>
<td>Specificity</td>
<td>90.57</td>
<td>99.33</td>
<td>90.43</td>
<td>90.53</td>
<td>91.04</td>
</tr>
<tr>
<td></td>
<td>F1-Score</td>
<td>95.32</td>
<td>99.54</td>
<td>95.04</td>
<td>90.19</td>
<td>95.41</td>
</tr>
<tr>
<td></td>
<td>GM</td>
<td>95.17</td>
<td>99.65</td>
<td>94.86</td>
<td>90.93</td>
<td>95.42</td>
</tr>
<tr>
<td>Wine</td>
<td>Accuracy</td>
<td>93.20</td>
<td>97.00</td>
<td>92.50</td>
<td>89.90</td>
<td>94.20</td>
</tr>
<tr>
<td></td>
<td>Sensitivity</td>
<td>91.54</td>
<td>99.03</td>
<td>90.03</td>
<td>88.76</td>
<td>93.41</td>
</tr>
<tr>
<td></td>
<td>Specificity</td>
<td>92.68</td>
<td>94.64</td>
<td>91.89</td>
<td>88.14</td>
<td>94.32</td>
</tr>
<tr>
<td></td>
<td>F1-Score</td>
<td>92.84</td>
<td>94.54</td>
<td>90.95</td>
<td>89.64</td>
<td>94.12</td>
</tr>
<tr>
<td></td>
<td>GM</td>
<td>92.11</td>
<td>96.76</td>
<td>89.57</td>
<td>88.45</td>
<td>93.86</td>
</tr>
<tr>
<td>WDBC</td>
<td>Accuracy</td>
<td>94.40</td>
<td>96.10</td>
<td>92.80</td>
<td>88.70</td>
<td>93.60</td>
</tr>
<tr>
<td></td>
<td>Sensitivity</td>
<td>96.30</td>
<td>96.43</td>
<td>92.86</td>
<td>86.21</td>
<td>93.31</td>
</tr>
<tr>
<td></td>
<td>Specificity</td>
<td>93.33</td>
<td>96.55</td>
<td>90.84</td>
<td>89.29</td>
<td>93.74</td>
</tr>
<tr>
<td></td>
<td>F1-Score</td>
<td>94.55</td>
<td>96.43</td>
<td>89.86</td>
<td>87.72</td>
<td>92.86</td>
</tr>
<tr>
<td></td>
<td>GM</td>
<td>94.80</td>
<td>96.49</td>
<td>91.42</td>
<td>87.74</td>
<td>93.53</td>
</tr>
</tbody>
</table>

### Appendix

The value of $l$ is calculated as follows:

$$i_{\text{LowMedium}}(x) = \begin{cases} 0 & \text{if } x \leq a \text{ or } x \geq c \\ x - a & \text{if } a < x \leq b \\ \frac{c - x}{m} & \text{if } b < x < c \\ \end{cases}$$ (5)

and

$$i_{\text{MediumHigh}}(x) = \begin{cases} 0 & \text{if } x \leq e \text{ or } x \geq f \\ x - d & \text{if } d < x < e \\ \frac{f - x}{m} & \text{if } e < x < f \\ \end{cases}$$ (6)

where $m = \max - \min$, $a = \min + 2m - \frac{m}{4}$, $b = \min + 2m - \frac{m}{4}$, $c = \min + 2m$, $d = \min + 3m$, $e = \min + 3m - \frac{m}{4}$, and $f = \min + 4m$.

The value of $T$ is calculated according to Eqs. (7–9).

$$t_{\text{Low}}(x) = \begin{cases} 1 & \text{if } x \leq a \\ \frac{b - x}{3m} & \text{if } a < x < b \\ 0 & \text{if } x \geq b \\ \end{cases}$$ (7)

$$t_{\text{Medium}}(x) = \begin{cases} 0 & \text{if } x < b \text{ or } x > e \\ \frac{x - b}{m} & \text{if } b \leq x < c \\ \frac{e - x}{4m} & \text{if } d < x \leq e \\ \frac{4 - x}{m} & \text{if } c \leq x \leq d \\ \end{cases}$$ (8)

$$t_{\text{High}}(x) = \begin{cases} 0 & \text{if } x < e \\ \frac{x - e}{3m} & \text{if } e \leq x < f \\ \frac{4 - e}{m} & \text{if } x \geq f \\ \end{cases}$$ (9)

Fig. 7. Visualization of how the indeterminacy ($l$) value is calculated.

Fig. 8 shows that the value of $T$ decreases when there is an intersection between the classes, and the value of $T$ reached to the
maximum value when there is no any intersection between the classes.

Similarly, value of \( F \) is calculated as follows:

\[
f_{\text{Low}}(x) = \begin{cases} 
0 & \text{if } x < a \\
\frac{x - a}{\frac{2}{m}} & \text{if } a \leq x \leq b \\
\frac{x}{\frac{2}{b}} & \text{if } x > b
\end{cases}
\]

(10)

\[
f_{\text{Medium}}(x) = \begin{cases} 
0 & \text{if } b < x < c \\
\frac{b - x}{\frac{2}{m}} & \text{if } a \leq x \leq b \\
\frac{x - c}{\frac{2}{c}} & \text{if } c < x \leq d \\
\frac{x}{\frac{2}{d}} & \text{if } x < a \text{ or } x > d
\end{cases}
\]

(11)

\[
f_{\text{High}}(x) = \begin{cases} 
0 & \text{if } x > d \\
\frac{d - x}{\frac{2}{d}} & \text{if } c \leq x \leq d \\
\frac{x}{\frac{2}{c}} & \text{if } x < c
\end{cases}
\]

(12)

where \( a = \min + 2m - \frac{m}{4} \), \( b = \min + 2m \), \( c = \min + 3m \), and \( d = \min + 3m + \frac{m}{4} \). Fig. 9 can be used for calculating the value of \( f \). As shown, when \( x < n \) the value of \( f_{\text{Low}} \) is zero while the value of \( f_{\text{Low}} \) is one (see Fig. 8). Moreover, if \( x > b \), the values of \( f_{\text{Low}} \) and \( f_{\text{Low}} \) are zero and one, respectively.

CredII authorship contribution statement

Sameh H. Basha: Made and design the analysis for the data, design the experiments, wrote the paper, and revised the paper.

Alaa Tharwat: Wrote the paper, analyzed the data, and revised the paper.

Areeg Abdalla: Analyzed the data, wrote and revised the paper.

Aboul Ella Hassanien: Helped in design the proposed algorithm, and revised the paper.

References


