Expert Systems With Applications 121 (2019) 142–157

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

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

Sameh H. Basha[a], Alaa Tharwat[b,d,1,*], Areeg Abdalla[a], Aboul Ella Hassanien[c,d]

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

ARTICLE INFO

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

ABSTRACT

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 utilized 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 (Prival, 2001), DEREK (Woo et al., 1995), Case (Klopman, 1984) and Multicase (Klopman, 1992). Tharwat, Gabel,

* Corresponding author.
E-mail addresses: Samehbasha@Sci.cu.edu.eg (S.H. Basha), aothman@fb2.fra-uas.de (A. Tharwat), Areeg@Sci.cu.edu.eg (A. Abdalla).
URL: http://www.egyptscience.net (A.E. Hassanien)
[1] My present address is Faculty of Computer Science and Engineering, Frankfurt University of Applied Sciences, Frankfurt am Main, Germany.

https://doi.org/10.1016/j.eswa.2018.12.014
0957-4174/© 2018 Elsevier Ltd. All rights reserved.

and Hassanien (2017), Tharwat, Moemen, and Hassanien (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 tradeoffs. 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

Neutrosophy 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 $<$Anti–A$>$, and (3) a spectrum of "neutralities" $<$Neut–A$>$. The $<$Neut–A$>$ and $<$Anti–A$>$ together are referred to as $<$Non–A$>$ (Wang et al., 2005). Both $<$Anti–A$>$ and $<$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 $\mu_c(x)$ that determines its membership value or score for each class (c) (Amo, Montero, Biging, & Cutello, 2004; Ansari, Biswas, & Aggarwal, 2013). For example, let $C = 3$, i.e. three classes, and

Table 1
Dataset description.

Feature no.	Name	Feature no.	Name
1	Total molecular weight	17	Electron negative atoms
2	Molecular weight	18	Stereo centers
3	Absolute weight	19	Rotatable bonds
4	cLogP (Octanol/Water, partition coefficient)	20	Rings
5	cLogS (Aqueous solubility)	21	Aromatic rings
6	H-Acceptors (Hydrogen bond Acceptor)	22	Aromatic atoms
7	H-Donors (Hydrogen bond donor)	23	sp3-Atoms
8	Total surface area	24	Symmetric atoms
9	Polar surface area	25	Amides (acid amide)
10	Druglikeness	26	Amines
11	Molecular shape index	27	AlkylAmines
12	Molecular flexibility	28	Aromatic amines
13	Molecular complexity	29	Aromatic nitrogen
14	Non hydrogen atoms	30	Basic nitrogen
15	Non-Carbon/Hydrogen atoms	31	Acidic oxygen
16	Metal atoms		

Table 2
The number of positive and negative samples and the imbalance ratio for each toxic effect in our dataset.

Toxic effect	# Samples in positive class	# Samples in negative class	Imbalance ratio
Mutagenic effect	90=16.28%	463=83.73%	5.14
Tumorigenic effect	90=16.28%	463=83.73%	5.14
Reproductive effect	187=33.82%	366=66.18%	1.96
Irritant effect	67=12.16%	486=87.88%	7.25

$\mu(x) = \{0.2, 0.5, 0.3\}$. 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 the concept of all 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).

3.1.1. 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, 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 $]^-0, 1^+[$[2] 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, $^-0 \leq supT_A(x) + supI_A(x) + supF_A(x) \leq 3^+$, where $supT_A$, $supI_A$ and $supF_A$ 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 $\bar{A}$ and defined by Smarandache (2003), Wang et al. (2005), Arora et al. (2011):

$$T_{\bar{A}}(x) = \{1^+\} - T_A(x),$$
$$I_{\bar{A}}(x) = \{1^+\} - I_A(x),$$
$$F_{\bar{A}}(x) = \{1^+\} - F_A(x).$$

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

$$T_C(x) = T_A(x) + T_B(x) - T_A(x) \times T_B(x),$$
$$I_C(x) = I_A(x) + I_B(x) - I_A(x) \times I_B(x),$$
$$F_C(x) = F_A(x) + F_B(x) - F_A(x) \times F_B(x).$$

  for all x in X.
- **Intersection**: The intersection of two neutrosophic sets A and B denoted by $C = A \cap B$ and defined by Smarandache (2003), Wang et al. (2005), Arora et al. (2011):

$$T_C(x) = T_A(x) \times T_B(x),$$
$$I_C(x) = I_A(x) \times I_B(x),$$
$$F_C(x) = F_A(x) \times F_B(x).$$

  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)

$$infT_A(x) \leq infT_B(x); supT_A(x) \leq supT_B(x),$$
$$infI_A(x) \leq infI_B(x); supI_A(x) \leq supI_B(x),$$

[2] The notation $]^-0, 1^+[$ represents the non-standard interval where $^-0 = 0 - \epsilon$ and $1^+ = 1 + \epsilon$, where ϵ is an infinitesimal number which is a number that is larger than each negative real number and is smaller than each positive real number, i.e. ϵ is an infinitesimal if $|\epsilon| < \frac{1}{n}$ for all $n \in N$ (Robinson, 2003).

$$infF_A(x) \leq infF_B(x); supF_A(x) \leq supF_B(x).$$

where $supT_A$, $supI_A$ and $supF_A$ represent the supremum for the T_A, I_A and F_A, respectively, and $infT_A$, $infI_A$ and $infF_A$ are the infimum of the T_A, I_A and F_A, respectively, and $x \in X$.

3.1.2. Neutrosophic logic

Neutrosophic 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). Neutrosophic 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 $]^{-}0, 1^{+}[$ where $]^{-}0, 1^{+}[$ is a non-standard unit interval (Ansari et al., 2013; Robinson, 2003). T, I, and F are called neutrosophic components, and these components represent the truth, indeterminacy, and falsehood values respectively in studying neutrosophy, neutrosophic logic, neutrosophic set, neutrosophic probability, neutrosophic statistics (Ashbacher, 2002). In real world applications, it is easier to use standard real interval [0,1] for T, I, and F instead of the non-standard unit interval $]^{-}0, 1^{+}[$ (Ansari et al., 2013; Basha, Sahlol, El Baz, & Hassanien, 2017).

The sets T, I, and F are not necessarily intervals, but may be any real sub-unitary 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, and F are subsets, but dynamically the components T, I, and F are set-valued vector functions/operators depending on many parameters, such as: time, space, etc. (Smarandache, 2003).

3.2. Neutrosophic rule-based classification system

In 2013 Ansari et al. (2013) presented a neutrosophic 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 Neutrosophic Rule-based Classification System (NRCS) uses Neutrosophic Logic (NL) for generalizing the fuzzy rule-based classification system. The antecedents and consequents of the "IF-THEN" rules in the NRCS are neutrosophic logic statements, instead of fuzzy logic ones. The NRCS has three stages:

1. **Neutrosophication:** construction of the neutrosophic knowledge-base by converting crisp inputs using the neutrosophic three membership functions: truth-membership, falsity-membership, and indeterminacy-membership.
2. **Inference Engine:** The KB and neutrosophic "IF-THEN" rules are applied to get a neutrosophic output,
3. **Deneutrosophication:** Converts the neutrosophic output of the previous step back to a crisp value using three functions analogous to the ones used by the neutrosophication.

The used knowledge base (KB) stores the available knowledge in the form of neutrosophic "IF-THEN" rules, and then the KB captures the neutrosophic rule semantics using neutrosophic sets. Fig. 1 shows the NRCS consisting of four phases: information extraction phase, neutrosophication 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. Neutrosophication 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 neutrosophic 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 ($X_{training}$) and testing data ($X_{testing}$). In this phase, neutrosophic rules are generated from the training and testing data. The training rules are denoted by $R_{training} = \{r^1_{tr}, r^2_{tr}, \ldots, r^{n_{tr}}_{tr}\}$, where r^i_{tr} is the rule for the i^{th} training sample and n_{tr} is the number of training samples. Similarly, the testing rules are denoted by $R_{testing} = \{r^1_{ts}, r^2_{ts}, \ldots, r^{n_{ts}}_{ts}\}$, where r^i_{ts} is the rule for the i^{th} testing sample and n_{ts} is the number of testing samples. In NRCS, the attribute in each neutrosophic rule has three components $< T, I, F >$.

3.2.4. Classification phase

In this phase, for each testing rule ($r^i_{ts} \in R_{testing}$), the Euclidean distance between a testing rule and all training rules ($R_{training}$) 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 neutrosophic 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, (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 neutrosophic class indicating 100% belongingness to class I in the range between 0 and a, and 100%

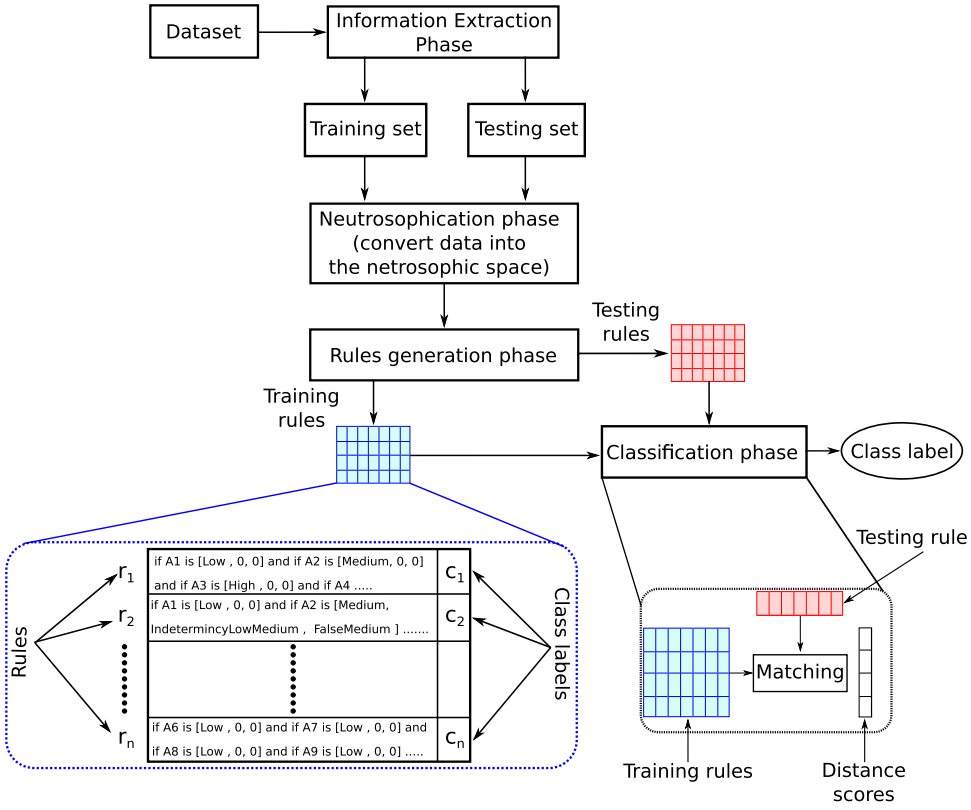

Fig. 1. Block diagram of the proposed NRCS model.

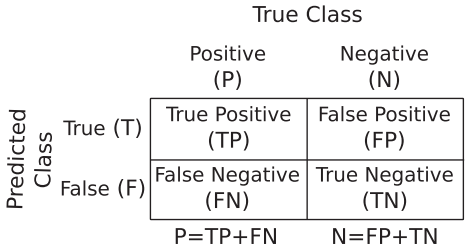

Fig. 2. An illustrative example of the 2×2 confusion matrix.

Table 3
The data for our illustrative example.

f_1	f_2	Class label (c)
0.2	5	A
2.5	6	A
3.2	3	A
0.5	2	A
1	3	A
3.3	4	B
3.5	5	B
4.5	5	B
6	2	B
6.2	6	B

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 (f_1 and f_2).
- The minimum of the first and second attributes are 0.2 and 2, respectively.
- The maximum of the first and second attribute are 6.2 and 6, respectively.
- There are two classes: class A and class B.

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: $< t_{low}, t_{Medium}, t_{High} >$, $< i_{Low}, i_{Medium} >$, $< f_{Low}, f_{Medium}, f_{High} >$. 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.

Value	t_{Low}	t_{Medium}	t_{High}	$i_{LowMedium}$	$i_{MediumHigh}$	f_{Low}	f_{Medium}	f_{High}
0.2	1.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0
5.0	0.0	0.0	0.667	0.0	0.0	1.0	1.0	0.0
2.5	0.0	0.667	0.0	0.334	0.0	0.833	0.166	1.0
6.0	0.0	0.0	1.0	0.0	0.0	1.0	1.0	0.0
3.2	0.0	1.0	0.0	0.0	0.0	1.0	0.0	1.0
3.0	0.667	0.0	0.0	0.0	0.0	0.0	1.0	1.0

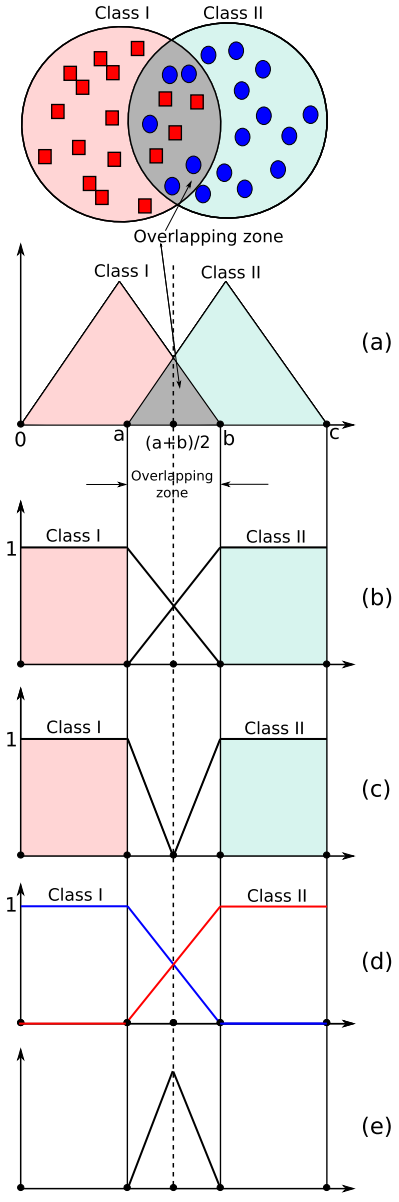

Fig. 3. A comparison between the fuzzy classifier and the NRCS classifier. (a) triangular membership function with two classes, (b) fuzzy classifier, (c) neutrosophic truth component, (d) neutrosophic falsity component, (e) neutrosophic indeterminacy component.

class *A* (see Table 3) is converted to $<1.0, 0.0, 0.0>$, $<0.0, 0.0>$, $<0.0, 1.0, 1.0>$ [3]

[3] More details about how to calculate how to calculate the values of *t*, *i*, and *f* are in the Appendix.

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:

$$\begin{aligned}
&[1, 0, 0][3, 0, 0] \rightarrow \text{class A}\\
&[2, 4, 7][3, 0, 0] \rightarrow \text{class A}\\
&[2, 0, 0][1, 0, 0] \rightarrow \text{class A}\\
&[2, 0, 0][2, 0, 0] \rightarrow \text{class B}\\
&[2, 0, 0][3, 0, 0] \rightarrow \text{class B}
\end{aligned} \tag{1}$$

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:

$$\begin{aligned}
&[1, 0, 0][1, 0, 0]\\
&[1, 0, 0][1, 0, 0]\\
&[3, 0, 0][1, 0, 0]\\
&[3, 0, 0][1, 0, 0]\\
&[3, 0, 0][3, 0, 0]
\end{aligned} \tag{2}$$

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

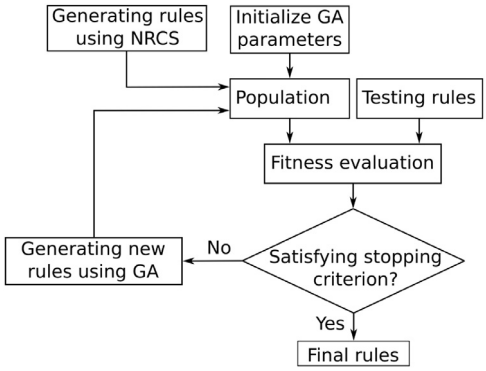

Fig. 4. Block diagram of the proposed GNRCS model.

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:

$$min\ \mathcal{F} = \frac{N_e}{N} \tag{3}$$

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, such as 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; Chen, Zhu, & Xu, 2015; Pawlak, 1982; Wang, Yang, Teng, Xia, & Jensen, 2007).

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

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

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 $\underline{P}X = \{x \in U | [x]_P \subseteq X\}$ and $\overline{P}X = \{x \in U | [x]_P \cap X \neq \phi\}$, respectively.

Let $P1$, $P2 \subseteq A$ be equivalent relations over U, the positive, negative, and boundary regions are defined, respectively, as follows: $POS_{P1}(P2) = \bigcup_{x \in U/P2} \underline{P1}X$, $NEG_{P1}(P2) = U - \bigcup_{x \in U/P2} \overline{P1}X$, $BND_{P1}(P2) = \bigcup_{x \in U/P2} \overline{P1}X - \bigcup_{x \in U/P2} \underline{P1}X$, where $POS_{P1}(P2)$ indicates the positive region of the relation $U/P2$ with respect to P, $NEG_{P1}(P2)$ represents the negative region, and $BND_{P1}(P2)$ 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, $k = \gamma(Q) = \frac{|POS_P(Q)|}{|U|}$, 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 $0 \leq k \leq 1$, 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 $\gamma_R(D) = \gamma_C(D)$, where R is the minimal subset if $\gamma_{R-a}(D) \neq \gamma_R(D), \forall a \in R$. 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 Reduct Feature Selection (QRFS) (Jensen & Shen, 2003), Discernibility 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 QRFS 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 & Garcia, 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 follow, $x_{new} = x_i + r_{ij} \times \delta = x_i + (\hat{x_{ij}} - x_i) \times \delta$, where $x_i \in S_{\min}$ is one of the minority class samples, $\hat{x_{ij}}$ is one of the k-nearest samples for $x_i : \hat{x_{ij}} \in S_{\min}, j = 1, 2, \ldots, k$, 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 $\hat{x_{ij}}$ (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).

Table 5
The detailed settings.

Name	Detailed settings
Hardware	
CPU	Core (TM) i5-2400
Frequency	3.10 GHz
RAM	4G
Hard Drive	160 GB
Software	
Operating system	Windows 7
Language	Java (Version 8) with NetBeans IDE 8.0.2

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 GNRCS parameter were found to be: $N = 200$, $P_c = 0.9$, 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 5×10-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 is repeated five times. The results are the average of these 5×10 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 without any 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.

Metrics	Mutagenic effect				Tumorigenic effect				Irritant effect				Reproductive effect			
	NN	LDA	KNN	NRCS	NN	LDA	KNN	NRCS	NN	LDA	KNN	NRCS	NN	LDA	KNN	NRCS
Accuracy	81.82	83.64	76.36	87.27	76.36	81.81	74.55	85.46	83.64	85.46	76.36	87.27	78.18	74.55	70.91	78.18
Sensitivity	28.57	33.33	20.00	50.00	20.00	36.36	18.18	44.44	33.33	33.33	18.18	42.86	71.43	64.29	56.25	68.75
Specificity	89.58	89.80	88.89	91.48	88.89	93.18	88.6	93.48	93.48	91.84	90.91	93.75	80.49	78.05	76.92	82.05
F_1-Score	28.57	30.77	23.53	46.15	23.53	44.44	22.22	50.00	40.00	33.33	23.53	46.15	62.50	56.25	52.94	64.71
GM	50.59	54.71	42.16	67.76	42.16	58.21	40.14	64.46	55.18	55.33	40.66	63.39	75.82	70.83	65.78	75.11

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.

Metrics	Mutagenic effect				Tumorigenic effect				Irritant effect				Reproductive effect			
	Bag	Ada	RF	NRCS	Bag	Ada	RF	NRCS	Bag	Ada	RF	NRCS	Bag	Ada	RF	NRCS
Accuracy	84.50	84.00	86.64	87.27	84.10	83.56	85.00	85.64	88.00	86.45	87.00	87.27	73.00	64.50	76.28	78.18
Sensitivity	38.57	40.27	46.87	50.00	42.78	41.52	43.75	44.44	43.15	39.54	41.98	42.86	64.05	53.45	66.48	68.75
Specificity	87.54	88.47	90.25	91.48	91.25	90.75	93.17	93.48	93.95	92.75	93.56	93.75	78.56	75.25	81.34	82.05
F_1-Score	36.78	40.54	42.98	46.15	46.57	45.19	48.97	50.000	47.36	43.46	45.25	46.15	60.36	58.61	62.48	64.71
GM	57.89	60.57	64.67	67.76	90.48	88.97	92.97	64.46	64.05	59.97	62.74	63.39	69.54	63.25	73.46	75.11

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 this experiment in terms of accuracy, sensitivity, specificity, F_1-score, and GM[4] 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 F_1-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 obtains 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 obtains also competitive results. And it is a natural result - after using the indeterminacy term in the neutrosophic logic - to have the NRCS determines the more significant, neutral and non-significant features without using 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. GNRCS

The aim of using the hybrid classification system (GNRCS) 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 GNRCS model represent the best results obtained.

From Table 8 it can be noted that in terms of accuracy, sensitivity, F_1-Score and GM metrics, the proposed GNRCS algorithm outperformed the NRCS algorithm in most cases. Moreover, in terms of specificity, the GNRCS and NRCS algorithms obtained competitive results. In other words, the GNRCS algorithm improved the sensitivity more than the specificity. This is because GA in the GNRCS algorithm searches for the optimal or near optimal rules and removes redundant rules; hence, the GA refines the generated neutrosophic 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 GNRCS 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.

Metrics	Mutagenic effect		Tumorigenic effect		Irritant effect		Reproductive effect	
	NRCS	GNRCS	NRCS	GNRCS	NRCS	GNRCS	NRCS	GNRCS
Accuracy	87.27	89.09	85.46	87.27	87.27	89.09	78.18	80.00
Sensitivity	50.00	57.14	44.44	50.00	42.86	50.00	68.75	70.59
Specificity	91.48	93.75	93.48	93.62	93.75	93.88	82.05	84.21
F_1-Score	46.15	57.14	50.00	53.33	46.15	50.00	64.71	68.67
GM	67.76	73.19	64.46	68.42	63.39	68.51	75.11	77.10

Table 9
Number of training rules of the proposed models (NRCS and GNRCS) with all datasets.

	Mutagenic effect	Tumorigenic effect	irritant effect	Reproductive effect
# NRCS rules	304	282	269	279
# GNRCS rules	277	229	267	250

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.

Classifier	Metrics	Orig.	RUS	ROS	SMOTE
DA+SVM (Tharwat, Gabel et al., 2017)	Sensitivity	48.00	54.19	86.49	89.43
	Specificity	86.60	75.43	82.46	86.55
	GM	64.47	63.93	84.45	88.00
WOA+SVM (Tharwat, Moemen et al., 2017)	Sensitivity	50.00	57.23	85.43	89.74
	Specificity	88.50	76.84	83.24	87.27
	GM	66.52	66.31	84.33	88.50
NRCS	Sensitivity	50.00	55.56	88.64	91.11
	Specificity	91.48	77.78	85.71	89.58
	GM	67.76	65.73	87.16	90.34
GNRCS	Sensitivity	57.14	62.50	86.96	91.36
	Specificity	93.75	80.00	87.23	91.49
	GM	73.19	70.70	87.10	91.40

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.

Classifier	Metrics	Orig.	RUS	ROS	SMOTE
DA+SVM (Tharwat, Gabel et al., 2017)	Sensitivity	42.41	60.74	90.84	90.05
	Specificity	91.62	67.67	85.66	86.57
	GM	62.33	64.11	88.21	88.29
WOA+SVM (Tharwat, Moemen et al., 2017)	Sensitivity	46.24	60.49	89.28	90.49
	Specificity	94.65	68.84	84.54	87.64
	GM	66.16	64.53	86.88	89.05
NRCS	Sensitivity	44.44	62.50	91.11	91.30
	Specificity	93.48	70.00	87.50	89.36
	GM	64.46	66.14	89.29	90.33
GNRCS	Sensitivity	50.00	66.67	93.33	91.49
	Specificity	93.62	77.78	89.58	91.30
	GM	68.42	72.01	91.44	91.40

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 inTables 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

Table 12
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 reproductive effect.

Classifier	Metrics	Orig.	RUS	ROS	SMOTE
DA+SVM (Tharwat, Gabel et al., 2017)	Sensitivity	64.58	68.21	78.24	81.73
	Specificity	81.33	67.88	80.27	86.54
	GM	72.47	68.05	79.25	84.10
WOA+SVM (Tharwat, Moemen et al., 2017)	Sensitivity	63.42	69.21	79.93	82.91
	Specificity	79.98	67.31	80.54	87.00
	GM	71.22	68.25	80.23	84.93
NRCS	Sensitivity	68.75	70.59	81.68	84.21
	Specificity	82.05	70.00	83.78	88.89
	GM	75.11	70.29	82.42	86.52
GNRCS	Sensitivity	70.59	77.78	86.49	87.18
	Specificity	84.21	78.95	89.19	94.29
	GM	77.10	78.36	87.83	90.66

Table 13
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 irritant effect.

Classifier	Metrics	Orig.	RUS	ROS	SMOTE
DA+SVM (Tharwat, Gabel et al., 2017)	Sensitivity	40.19	38.67	80.05	81.32
	Specificity	91.48	49.11	80.23	84.28
	GM	60.64	43.58	80.14	82.79
WOA+SVM (Tharwat, Moemen et al., 2017)	Sensitivity	41.28	39.18	79.94	84.63
	Specificity	90.87	47.28	82.41	83.24
	GM	61.25	43.04	81.16	83.93
NRCS	Sensitivity	42.86	42.86	81.63	83.67
	Specificity	93.75	50.00	83.33	85.42
	GM	63.39	46.29	82.48	84.54
GNRCS	Sensitivity	50.00	50.00	85.42	89.36
	Specificity	93.88	60.00	85.71	87.76
	GM	68.51	54.78	85.57	88.56

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 were 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, $S_{W_i} = \sum_{j=1}^{n_i} (x_{ij} - \mu_i)(x_{ij} - \mu_i)^T$, where x_{ij} is the jth sample in the ith class, S_{W_i} is the within-class variance of the ith class, n_i is the number of samples in the class i, and μ_i is the mean of the ith class. The total within-class variance is defined as follows, $S_W = \sum_{i=1}^{C} S_{W_i}$, where C is the number of classes

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.

Metrics	NRCS				GNRCS			
	Orig.	DMFS	EBFS	QRFS	Orig.	DMFS	EBFS	QRFS
Sensitivity	91.11	91.30	93.33	93.33	91.36	91.11	93.38	93.48
Specificity	89.58	91.49	91.62	91.67	91.49	89.58	91.67	93.62
GM	90.34	91.40	92.50	92.50	91.40	90.34	92.50	93.55

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.

Metrics	NRCS				GNRCS			
	Orig.	DMFS	EBFS	QRFS	Orig.	DMFS	EBFS	QRFS
Sensitivity	91.30	93.33	89.58	93.48	91.49	89.30	93.33	93.62
Specificity	89.36	91.67	93.48	93.75	91.30	91.49	91.67	95.75
GM	90.33	92.50	91.51	93.61	91.40	90.42	92.50	94.68

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.

Metrics	NRCS				GNRCS			
	Orig.	DMFS	EBFS	QRFS	Orig.	DMFS	EBFS	QRFS
Sensitivity	84.21	86.62	86.84	89.47	87.18	86.49	87.18	89.74
Specificity	88.89	91.43	91.67	94.44	94.29	89.19	94.29	97.14
GM	86.52	87.96	89.22	91.93	90.66	87.83	90.66	93.37

(Tharwat, Gaber, Ibrahim, & Hassanien, 2017). 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 (μ_i) and the total mean of all classes (μ). 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 $O((N + logM)M^2)$, 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 GNRC 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 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.

Metrics	NRCS				GNRCS			
	Orig.	DMFS	EBFS	QRFS	Orig.	DMFS	EBFS	QRFS
Sensitivity	83.67	85.71	89.36	89.80	89.36	89.36	89.80	91.84
Specificity	85.42	87.50	88.00	91.67	87.76	88.00	91.49	93.62
GM	84.54	86.60	88.68	90.73	88.56	88.68	90.64	92.72

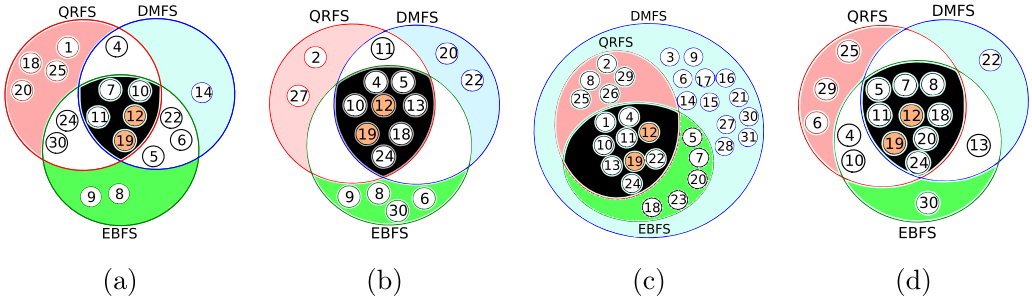

Fig. 5. Selected features of the three feature selection methods. (a) Mutagenic, (b) Tumorigenic, (c) Reproductive, (d) Irritant.

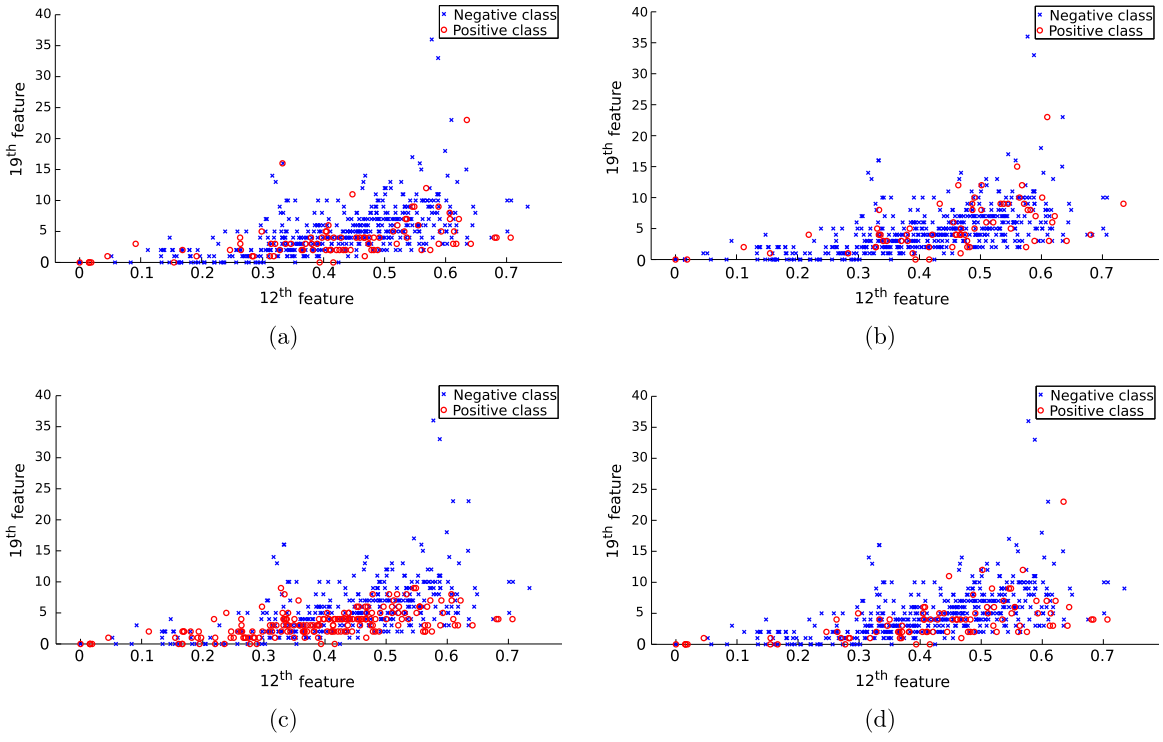

Fig. 6. Scatter plot for all datasets (a) Tumorigenic, (b) Irritant, (c) Reproductive, (d) Mutagenic. The scatter plot demonstrates the values of two of the most important selected features (12th and 19th features) which are highlighted in Fig. 5.

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-

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

Dataset	Metric	NRCS	GNRCS	MLP	*k*-NN	LDA
Iris	Accuracy	96.40	99.30	95.40	90.20	96.50
	Sensitivity	100.0	100.0	99.50	90.53	91.04
	Specificity	90.57	99.33	90.43	90.53	91.04
	F1-Score	95.32	99.54	95.04	90.19	95.41
	GM	95.17	99.65	94.86	90.93	95.42
Wine	Accuracy	93.20	97.00	92.50	89.90	94.20
	Sensitivity	91.54	99.03	90.03	88.76	93.41
	Specificity	92.68	94.64	91.89	88.14	94.32
	F1-Score	92.84	94.54	90.95	89.64	94.12
	GM	92.11	96.76	89.57	88.45	93.86
WDBC	Accuracy	94.40	96.10	92.80	88.70	93.60
	Sensitivity	96.30	96.43	92.86	86.21	93.31
	Specificity	93.33	96.55	90.84	89.29	93.74
	F1-Score	94.55	96.43	89.86	87.72	92.86
	GM	94.80	96.49	91.42	87.74	93.53

ing the QRFS algorithm which improved the classification performance and reduced the computational time. Overall, the proposed GNRCS with SMOTE algorithm 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 ($93.5 \pm 8\%$), specificity ($91.5 \pm 7.7\%$), and GM ($91.5 \pm 7.7\%$)) and in (Tharwat, Gabel et al., 2017)(sensitivity ($90.3 \pm 7.6\%$), specificity ($92.0 \pm 8.3\%$), and GM ($90.3 \pm 6.8\%$)). 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), Discernibility 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.

Appendix

The value of I is calculated as follows:

$$i_{LowMedium}(x) = \begin{cases} 0 & \text{if } x \le a \text{ or } x \ge c \\ \dfrac{x-a}{\frac{m}{4}} & \text{if } a < x \le b \\ \dfrac{c-x}{\frac{m}{4}} & \text{if } b < x < c \end{cases} \tag{5}$$

and

$$i_{MediumHigh}(x) = \begin{cases} 0 & \text{if } x \le e \text{ or } x \ge f \\ \dfrac{x-d}{\frac{m}{4}} & \text{if } d < x < e \\ \dfrac{f-x}{\frac{m}{4}} & \text{if } e < x < f \end{cases} \tag{6}$$

where $m = \frac{max-min}{5}$, $a = min + 2m - \frac{m}{2}$, $b = min + 2m - \frac{m}{4}$, $c = min + 2m$, $d = min + 3m$, $e = min + 3m + \frac{m}{4}$, $f = min + 3m + \frac{m}{2}$, *min* and *max* represent the minimum and maximum of the attribute, respectively. The points *a, b, c, d, e*, and *f* are shown in Fig. 7. As shown, the triangles represent the intersections between different classes. Moreover, the value of I is zero when the input value is outside the triangle (see Eqs. (5) and (6)). This means that if there is no intersection between the classes, the value of I is zero; otherwise, $I > 0$. Additionally, the value of I increases by increasing the intersection between classes.

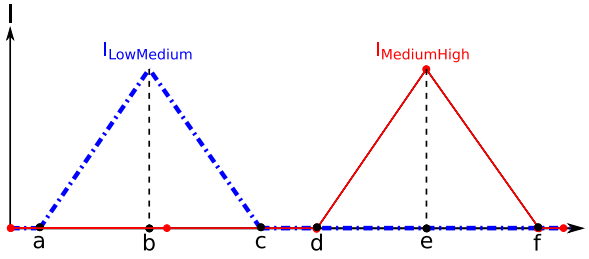

Fig. 7. Visualization of how the indeterminacy (I) value is calculated.

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

$$t_{Low}(x) = \begin{cases} 1 & \text{if } x \le a \\ \dfrac{b-x}{\frac{3m}{4}} & \text{if } a < x < b \\ 0 & \text{if } x \ge b \end{cases} \tag{7}$$

$$t_{Medium}(x) = \begin{cases} 0 & \text{if } x < b \text{ or } x > e \\ \dfrac{x-b}{\frac{m}{4}} & \text{if } b \le x < c \\ \dfrac{e-x}{\frac{m}{4}} & \text{if } d < x \le e \\ 1 & \text{if } c \le x \le d \end{cases} \tag{8}$$

$$t_{High}(x) = \begin{cases} 0 & \text{if } x < e \\ \dfrac{x-e}{\frac{3m}{4}} & \text{if } e \le x < f \\ 1 & \text{if } x \ge f \end{cases} \tag{9}$$

where $m = \frac{max-min}{5}$, $a = min + m$, $b = min + 2m - \frac{m}{4}$, $c = min + 2m$, $d = min + 3m$, $e = min + 3m + \frac{m}{4}$, and $f = min + 4m$. 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.

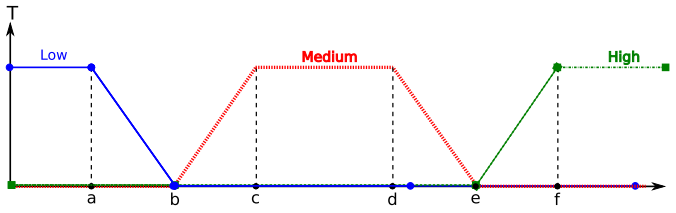

Fig. 8. Visualization of how the truth (T) value is calculated.

Similarly, value of F is calculated as follows:

$$f_{Low}(x) = \begin{cases} 0 & \text{if } x < a \\ \frac{x-a}{\frac{m}{2}} & \text{if } a \leq x \leq b \\ 1 & \text{if } x > b \end{cases} \quad (10)$$

$$f_{Medium}(x) = \begin{cases} 0 & \text{if } b < x < c \\ \frac{b-x}{\frac{m}{2}} & \text{if } a \leq x \leq b \\ \frac{x-c}{\frac{m}{2}} & \text{if } c \leq x \leq d \\ 1 & \text{if } x < a \text{ or } x > d \end{cases} \quad (11)$$

$$f_{High}(x) = \begin{cases} 0 & \text{if } x > d \\ \frac{d-x}{\frac{m}{2}} & \text{if } c \leq x \leq d \\ 1 & \text{if } x < c \end{cases} \quad (12)$$

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

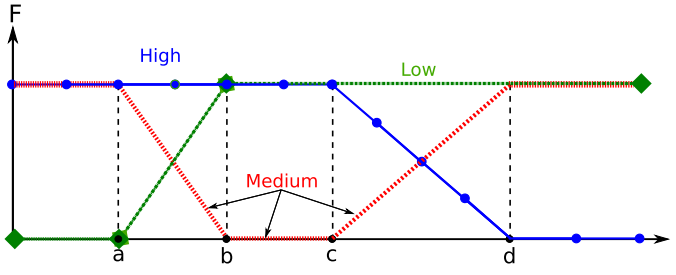

Fig. 9. Visualization of how the falsity (F) value is calculated.

Eqs. (5)–(12) are used to calculate the values of t_{Low}, t_{Medium}, t_{High}, $i_{LowMedium}$, $i_{MediumHigh}$, f_{Low}, f_{Medium}, and f_{High} as shown in Table 4. To generate a rule, the value of each feature (x) must be represented into only three values $< T, I, F >$, where:

- T is the maximum of t_{Low}, t_{Medium}, and t_{High},
- I is the maximum of $i_{LowMedium}$ and $i_{MediumHigh}$,
- F is the minimum of f_{Low}, f_{Medium}, and f_{High}.

In Eq. (1), the values 1, 2, 3, 4, 5, 6, 7, 8, and 9 represent t_{Low}, t_{Medium}, t_{High}, $i_{LowMedium}$, $i_{MediumHigh}$, f_{Low}, f_{Medium}, and f_{High}, respectively. Hence, instead of using symbols we used numbers.

CRediT 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

Al Iqbal, M. R. (2012). Rule extraction from ensemble methods using aggregated decision trees. In *International conference on neural information processing* (pp. 599–607). Springer.

Alblowi, S., Salama, A., & Eisa, M. (2013). New concepts of neutrosophic sets. *International Journal of Mathematics and Computer Applications Research (IJMCAR), 3*(4), 95–102.

Amo, A., Montero, J., Biging, G., & Cutello, V. (2004). Fuzzy classification systems. *European Journal of Operational Research, 156*(2), 495–507.

Ansari, A. Q., Biswas, R., & Aggarwal, S. (2013). Neutrosophic classifier: An extension of fuzzy classifer. *Applied Soft Computing, 13*(1), 563–573.

Arora, M., Biswas, R., & Pandy, U. (2011). Neutrosophic relational database decomposition. *International Journal of Advanced Computer Science and Applications, 2*(8), 121–125.

Ashbacher, C. (2002). *Introduction to neutrosophic logic*. Infinite Study.

Atanassov, K. T. (1989). More on intuitionistic fuzzy sets. *Fuzzy Sets and Systems, 33*(1), 37–45.

Basha, S. H., Abdalla, A. S., & Hassanien, A. E. (2016a). Gnrcs: hybrid classification system based on neutrosophic logic and genetic algorithm. In *Computer engineering conference (ICENCO), 2016 12th international* (pp. 53–58). IEEE.

Basha, S. H., Abdalla, A. S., & Hassanien, A. E. (2016b). Nrcs: Neutrosophic rule-based classification system. In *Proceedings of SAI intelligent systems conference* (pp. 627–639). Springer.

Basha, S. H., Sahlol, A. T., El Baz, S. M., & Hassanien, A. E. (2017). Neutrosophic rule-based prediction system for assessment of pollution on benthic foraminifera in burullus lagoon in egypt. In *12th international conference on computer engineering and systems (ICCES)* (pp. 663–668). IEEE.

Basha, S. H., Tharwat, A., Ahmed, K., & Hassanien, A. E. (2018). A predictive model for seminal quality using neutrosophic rule-based classification system. In *International conference on advanced intelligent systems and informatics* (pp. 495–504). Springer.

Blake, C. L. (1998). Uci repository of machine learning databases, irvine, university of California http://www.ics.uci.edu/~mlearn/MLRepository.html..

Breiman, L. (1996). Bagging predictors. *Machine Learning, 24*(2), 123–140.

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32.

Cao, D.-S., Yang, Y.-N., Zhao, J.-C., Yan, J., Liu, S., Hu, Q.-N., et al. (2012). Computer-aided prediction of toxicity with substructure pattern and random forest. *Journal of Chemometrics, 26*(1–2), 7–15.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 321–357.

Chen, Y., Miao, D., & Wang, R. (2010). A rough set approach to feature selection based on ant colony optimization. *Pattern Recognition Letters, 31*(3), 226–233.

Chen, Y., Zhu, Q., & Xu, H. (2015). Finding rough set reducts with fish swarm algorithm. *Knowledge-Based Systems, 81*, 22–29.

Elhoseny, M., Tharwat, A., & Hassanien, A. E. (2018). Bezier curve based path planning in a dynamic field using modified genetic algorithm. *Journal of Computational Science, 25*, 339–350.

Freund, Y., Schapire, R. E., et al. (1996). Experiments with a new boosting algorithm. In *Icml: 96* (pp. 148–156). Citeseer.

Friedman, J. H., Popescu, B. E., et al. (2008). Predictive learning via rule ensembles. *The Annals of Applied Statistics, 2*(3), 916–954.

Hassanien, A. E., Basha, S. H., & Abdalla, A. S. (2018). Generalization of fuzzy c-means based on neutrosophic logic. *Studies in Informatics and Control, 27*(1), 43–54.

He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering, 21*(9), 1263–1284.

Huang, R., Southall, N., Xia, M., Cho, M.-H., Jadhav, A., Nguyen, D.-T., et al. (2009). Weighted feature significance (wfs): A simple, interpretable model of compound toxicity based on the statistical enrichment of structural features. *Toxicological Sciences*, 1–33.

Inbarani, H. H., Azar, A. T., & Jothi, G. (2014). Supervised hybrid feature selection based on pso and rough sets for medical diagnosis. *Computer Methods and Programs in Biomedicine, 113*(1), 175–185.

Jensen, R., & Shen, Q. (2003). Finding rough set reducts with ant colony optimization. In *Proceedings of the 2003 UK workshop on computational intelligence: 1*.

Klopman, G. (1984). Artificial intelligence approach to structure-activity studies. computer automated structure evaluation of biological activity of organic molecules. *Journal of the American Chemical Society, 106*(24), 7315–7321.

Klopman, G. (1992). Multicase 1. A hierarchical computer automated structure evaluation program. *Quantitative Structure-Activity Relationships, 11*(2), 176–184.

von Korff, M., & Sander, T. (2006). Toxicity-indicating structural patterns. *Journal of Chemical Information and Modeling, 46*(2), 536–544.
López, V., Fernández, A., García, S., Palade, V., & Herrera, F. (2013). An insight into classification with imbalanced data: empirical results and current trends on using data intrinsic characteristics. *Information Sciences, 250*, 113–141.
Metawa, N., Hassan, M. K., & Elhoseny, M. (2017). Genetic algorithm based model for optimizing bank lending decisions. *Expert Systems with Applications, 80*, 75–82.
Pawlak, Z. (1982). Rough sets. *International Journal of Computer & Information Sciences, 11*(5), 341–356.
Pawlak, Z. (1991). *Rough sets: Theoretical aspects of reasoning about data*.
Pereira, M., Costa, V. S., Camacho, R., Fonseca, N. A., Simões, C., & Brito, R. M. (2009). Comparative study of classification algorithms using molecular descriptors in toxicological databases. In *Advances in bioinformatics and computational biology* (pp. 121–132). Springer.
Plewczynski, D. (2008). Tvscreen: Trend vector virtual screening of large commercial compounds collections. In *Proceedings of international conference on biocomputation, bioinformatics, and biomedical technologies, 2008. (BIOTECHNO'08)* (pp. 59–63).
Pritchard, J. F., Jurima-Romet, M., Reimer, M. L., Mortimer, E., Rolfe, B., & Cayen, M. N. (2003). Making better drugs: Decision gates in non-clinical drug development. *Journal of Nature Reviews Drug Discovery, 2*(7), 542–553.
Prival, M. J. (2001). Evaluation of the topkat system for predicting the carcinogenicity of chemicals. *Environmental and Molecular Mutagenesis, 37*(1), 55–69.
Robinson, A. (2003). Non-standard analysis. In *Mathematical logic in the 20th century* (pp. 385–393). World Scientific.
Sander, T., Freyss, J., von Korff, M., & Rufener, C. (2015). Datawarrior: An open-source program for chemistry aware data visualization and analysis. *Journal of Chemical Information and Modeling, 55*(2), 460–473.
Schapire, R. E., Freund, Y., Bartlett, P., Lee, W. S., et al. (1998). Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics, 26*(5), 1651–1686.
Smarandache, F. (2003). *A unifying field in logics: Neutrosophic logic. neutrosophy, neutrosophic set, neutrosophic probability: Neutrosophic logic: neutrosophy, neutrosophic set, neutrosophic probability*. Infinite Study.
Sun, Y., Wong, A. K., & Kamel, M. S. (2009). Classification of imbalanced data: A review. *International Journal of Pattern Recognition and Artificial Intelligence, 23*(04), 687–719.
Tharwat, A. (2016). Linear vs. quadratic discriminant analysis classifier: A tutorial. *International Journal of Applied Pattern Recognition, 3*(2), 145–180.
Tharwat, A. (2018). Classification assessment methods. *Applied Computing and Informatics*.
Tharwat, A., Gabel, T., & Hassanien, A. E. (2017a). Classification of toxicity effects of biotransformed hepatic drugs using optimized support vector machine. In *International conference on advanced intelligent systems and informatics* (pp. 161–170). Springer.
Tharwat, A., Gaber, T., Fouad, M. M., Snasel, V., & Hassanien, A. E. (2015). Towards an automated zebrafish-based toxicity test model using machine learning. In *Proceedings of international conference on Communications, management, and information technology (ICCMIT'2015): 65* (pp. 643–651). Procedia Computer Science
Tharwat, A., Gaber, T., Ibrahim, A., & Hassanien, A. E. (2017b). Linear discriminant analysis: A detailed tutorial. *AI Communications, 30*(2), 169–190.
Tharwat, A., Mahdi, H., Elhoseny, M., & Hassanien, A. E. (2018). Recognizing human activity in mobile crowdsensing environment using optimized k-nn algorithm. *Expert Systems With Applications, 107*, 32–44.
Tharwat, A., Moemen, Y. S., & Hassanien, A. E. (2017c). Classification of toxicity effects of biotransformed hepatic drugs using whale optimized support vector machines. *Journal of Biomedical Informatics, 68*, 132–149.
Trawiński, K., Cordón, O., & Quirin, A. (2011). On designing fuzzy rule-based multiclassification systems by combining furia with bagging and feature selection. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 19*(04), 589–633.
Turksen, I. B. (1986). Interval valued fuzzy sets based on normal forms. *Fuzzy Sets and Systems, 20*(2), 191–210.
Ulrich, R., & Friend, S. H. (2002). Toxicogenomics and drug discovery: Will new technologies help us produce better drugs. *Journal of Nature Reviews Drug Discovery, 1*(1), 84–88.
Wang, H., Smarandache, F., Sunderraman, R., & Zhang, Y.-Q. (2005). *Interval neutrosophic sets and logic: Theory and applications in computing: Theory and applications in computing*: 5. Infinite Study.
Wang, R., Miao, D., & Hu, G. (2006). Discernibility matrix based algorithm for reduction of attributes. In *Proceedings of the 2006 IEEE/WIC/ACMinternational conference on web intelligence and intelligent agent technology* (pp. 477–480). IEEE Computer Society.
Wang, X., Yang, J., Teng, X., Xia, W., & Jensen, R. (2007). Feature selection based on rough sets and particle swarm optimization. *Pattern Recognition Letters, 28*(4), 459–471.
Weiss, G. M., McCarthy, K., & Zabar, B. (2007). Cost-sensitive learning vs. sampling: Which is best for handling unbalanced classes with unequal error costs? *DMIN, 7*, 35–41.
Woo, Y.-T., Lai, D. Y., Argus, M. F., & Arcos, J. C. (1995). Development of structure-activity relationship rules for predicting carcinogenic potential of chemicals. *Toxicology Letters, 79*(1), 219–228.
Yamany, W., Tharwat, A., Hassanin, M. F., Gaber, T., Hassanien, A. E., & Kim, T.-H. (2015). A new multi-layer perceptrons trainer based on ant lion optimization algorithm. In *Fourth international conference on information science and industrial applications (ISI)* (pp. 40–45). IEEE.
Zadeh, L. A. (1996). Fuzzy sets. In *Fuzzy sets, fuzzy logic, and fuzzy systems: Selected papers by lotfi a zadeh* (pp. 394–432). World Scientific.