Chapter 3

Feature Selection Models and its Application

The intuitionistic fuzzy set is an interesting mathematical framework to deal with imprecise and/or imperfect information. An intuitionistic fuzzy set is inherently considered as an extension of fuzzy set (as proposed by Zadeh [168]), which is determined by a pair of membership and non-membership functions. It has been effectively applied for feature selection by eliminating redundant and irrelevant features. Various intuitionistic fuzzy set models proposed during the study are discussed in this chapter.

3.1 Divergence based Intuitionistic Fuzzy Rough Set Model

In this chapter, an intuitionistic fuzzy rough set model for feature selection is proposed using intuitionistic fuzzy divergence measure [128]. Intuitionistic fuzzy divergence measure expresses the extent to which two intuitionistic fuzzy sets differ from each other based on their membership values and at the same time similar to each other with respect to non-membership values. The properties of lower and upper approximations are also explored. Moreover, the necessary results on lower and upper approximations based on rough sets are extended for divergence measure based intuitionistic fuzzy rough sets and analogous results are presented.

Let $IF(U)$ be family of all intuitionistic fuzzy sets on U. An intuitionistic fuzzy divergence measure [111] given by $\delta : IF(U) \times IF(U) \implies \mathbb{R}$ satisfies the following properties for every $A, B \in IF(U)$:

- $\delta(A, B) = \delta(B, A)$
- $\delta(A, A) = 0$
- $\delta(A \cap B, B \cap C) \leq \delta(A, B), \forall C \in IF(U)$
- $\delta(A \cup B, B \cup C) \leq \delta(A, B), \forall C \in IF(U)$

which is more restrictive than similarity and dissimilarity measures [110].

If R is an intuitionistic fuzzy relation induced on the system, then the intuitionistic fuzzy lower and upper approximations (as defined in chapter (2.13)) reduces to following equations on simplification:

$$R \downarrow_A X = (R \downarrow_A \mu_X, R \uparrow_A \nu_X) \tag{3.1}$$

$$R \uparrow_A X = (R \uparrow_A \mu_X, R \downarrow_A \nu_X) \tag{3.2}$$

Clearly, membership function of intuitionistic fuzzy lower approximation is lower approximation of membership of X and non membership is upper approximation of non-membership of X. Dual proposition holds for upper approximation of X. Substituting the values of lower and upper approximations gives:

$$R \downarrow_A X(x) = (inf_{y \in U} max(1 - R(x,y), \mu_X(y)), sup_{y \in U} min(R(x,y), \nu_X(y))) \quad (3.3)$$

$$R \uparrow_A X(x) = (sup_{y \in U} min(R(x,y), \mu_X(y)), inf_{y \in U} max(1 - R(x,y), \nu_X(y))) \quad (3.4)$$

The value of divergence measure gives the dissimilarity between instance x and y, whereas $R(x,y)$ gives the extent of similarity between instances. Therefore, $R(x,y)$ could be replaced by $(1 - \delta(x,y))$. The intuitionistic fuzzy lower and upper approximation with respect to divergence measure δ is defined as follows:

$$\delta \downarrow_A X(x) = (inf_{y \in U} max(\delta(x,y), \mu_X(y)), sup_{y \in U} min(1 - \delta(x,y), \nu_X(y))) \quad (3.5)$$

$$\delta \uparrow_A X(x) = (sup_{y \in U} min(1 - \delta(x,y), \mu_X(y)), inf_{y \in U} max(\delta(x,y), \nu_X(y))) \quad (3.6)$$

The above defined approximations are intuitionistic fuzzy sets on U and could be asserted by the following proposition.

Proposition 3.1.1. The divergence based intuitionistic fuzzy lower and upper approximations are intuitionistic fuzzy set in U.

Proof. Since,

$$\mu_X(y), \nu_X(y), \delta(x,y) \in [0.1], \forall x, y \in U$$

Hence,

$$max(\delta(x,y), \mu_X(y)), min(1 - \delta(x,y), \nu_X(y)) \in [0,1], \forall x, y \in U$$

$$\implies \mu_{\delta \downarrow_A X}(x), \nu_{\delta \downarrow_A X}(x) \in [0,1], \forall x \in U$$

Similarly,

$$\mu_{\delta \uparrow_A X}(x), \nu_{\delta \uparrow_A X}(x) \in [0,1], \forall x \in U$$

Further, it can be shown by [69] that $0 \leq \mu_{\delta \downarrow_A X}(x) + \nu_{\delta \downarrow_A X}(x) \leq 1$ and $0 \leq \mu_{\delta \uparrow_A X}(x) + \nu_{\delta \uparrow_A X}(x) \leq 1 \implies \delta \downarrow_A X$ and $\delta \uparrow_A X$ are intuitionistic fuzzy sets. □

The various properties of proposed approximations are proved in the following theorems:

Theorem 3.1.1. $\delta \downarrow \phi = \phi = \delta \uparrow \phi$

Proof.

$$\begin{aligned}
&\mu_\phi(x) = 0, \delta(x,x) = 0, \forall x \in U\\
\implies &max(\delta(x,x), \mu_\phi(x)) = 0\\
\implies &inf_{y\in U} max(\delta(x,y), \mu_\phi(y)) = 0\\
&\nu_\phi(x) = 1, \forall x \in U\\
\implies &min(1-\delta(x,x), \nu_\phi(x)) = 1\\
\implies &sup_{y\in U} min(1-\delta(x,y), \nu_\phi(y)) = 1
\end{aligned}$$

Similarly,

$$\begin{aligned}
&min(1-\delta(x,x), \mu_\phi(y)) = 0\\
\implies &sup_{y\in U} min(1-\delta(x,y), \mu_\phi(y)) = 0
\end{aligned}$$

and

$$\begin{aligned}
&max(\delta(x,y), \nu_\phi(y)) = 1\\
\implies &inf_{y\in U} max(\delta(x,y), \nu_\phi(y)) = 1
\end{aligned}$$

Hence, $\delta \downarrow \phi = \phi = \delta \uparrow \phi$ □

Theorem 3.1.2. $\delta \downarrow U = U = \delta \uparrow U$

Proof.

$$\begin{aligned}
&\mu_U(x) = 1, \forall x \in U \implies max(\delta(x,y), \mu_U(y)) = 1\\
\implies &inf_{y\in U} max(\delta(x,y), \mu_U(y)) = 1\\
&\nu_U(x) = 0, \forall x \in U \implies min(1-\delta(x,y), \nu_U(y)) = 0\\
\implies &sup_{y\in U} min(1-\delta(x,y), \nu_U(y)) = 0
\end{aligned}$$

Since,

$$\delta(x,x) = 0, \forall x \in U \implies min(1-\delta(x,x), \mu_U(y)) = 1$$

$$\implies sup_{y\in U} min(1-\delta(x,y), \mu_U(y)) = 1$$

and

$$max(\delta(x,x), \nu_U(y)) = 0$$

$$\implies inf_{y\in U} max(\delta(x,y), \nu_U(y)) = 0$$

Hence, $\delta \downarrow U = U = \delta \uparrow U$ □

Theorem 3.1.3. $\delta \downarrow X \subseteq X \subseteq \delta \uparrow X$

Proof.

$$\delta(x,x) = 0, \forall x \in U \implies max(\delta(x,x), \mu_X(x)) = \mu_X(x), \forall x \in U$$

$$\implies inf_{y\in U} max(\delta(x,y), \mu_X(y)) \leq \mu_X(x), \forall x \in U$$

Also,

$$min(1-\delta(x,x), \nu_X(x)) = \nu_X(x), \forall x \in U$$

$$\implies sup_{y\in U} min(1-\delta(x,y), \nu_X(y)) \geq \nu_X(x), \forall x \in U$$

$$\implies \delta \downarrow X(x) \subseteq X(x), \forall x \in U$$

Similarly,

$$min(1-\delta(x,x), \mu_X(x)) = \mu_X(x), \forall x \in U$$

$$\implies sup_{y\in U} min(1-\delta(x,y), \mu_X(y)) \geq \mu_X(x), \forall x \in U$$

and

$$max(\delta(x,x), \nu_X(x)) = \nu_X(x), \forall x \in U$$

$$\implies inf_{y\in U} max(\delta(x,y), \nu_X(y)) \leq \nu_X(x), \forall x \in U$$

Hence, $\delta \uparrow X(x) \supseteq X(x) \implies \delta \downarrow X \subseteq X \subseteq \delta \uparrow X$ □

Theorem 3.1.4. $X \subseteq Y \implies \delta \downarrow X \subseteq \delta \downarrow Y$ and $\delta \uparrow X \subseteq \delta \uparrow Y$

Proof. As

$$X \subseteq Y \implies \mu_X(x) \leq \mu_Y(x)$$

and

$$\nu_X(x) \geq \nu_Y(x), \forall x \in U$$

So,

$$\begin{aligned} & max(\delta(x,y), \mu_X(y)) \leq max(\delta(x,y), \mu_Y(y)) \\ & min(1-\delta(x,y), \nu_X(y)) \geq min(1-\delta(x,y), \nu_Y(y)) \\ \implies & \delta \downarrow X(x) \subseteq \delta \downarrow Y(x) \end{aligned}$$

Similarly,

$$\begin{aligned} & min(1-\delta(x,y), \mu_X(y)) \leq min(1-\delta(x,y), \mu_Y(y)) \\ & max(\delta(x,y), \nu_X(y)) \geq max(\delta(x,y), \nu_Y(y)) \\ \implies & \delta \uparrow X(x) \subseteq \delta \uparrow Y(x) \end{aligned}$$

□

Theorem 3.1.5. $\delta \downarrow (\alpha,\beta) = (\alpha,\beta) = \delta \uparrow (\alpha,\beta), \forall (\alpha,\beta) \in [0,1]$

Proof. Using theorem 3.1.4

$$\delta \downarrow (\alpha,\beta) \subseteq (\alpha,\beta)$$

Also,

$$\begin{aligned} & (\alpha,\beta)(y) = (\alpha,\beta), \forall y \in U \\ \implies & max(\delta(x,y), \mu_{(\alpha,\beta)}(y)) \geq \alpha \\ & min(1-\delta(x,y), \nu_{(\alpha,\beta)}(y)) \leq \beta, \\ \implies & \delta \downarrow (\alpha,\beta)(x) = (inf_{y \in U} max(\delta(x,y), \mu_{(\alpha,\beta)}(y)), sup_{y \in U} min(1-\delta(x,y), \mu_{(\alpha,\beta)}(y))) \\ & \geq (\alpha,\beta)(x) \\ \implies & \delta \downarrow (\alpha,\beta) \supseteq (\alpha,\beta) \\ \implies & \delta \downarrow (\alpha,\beta) = (\alpha,\beta) \end{aligned}$$

Similarly,

$$
\begin{aligned}
& (\alpha, \beta) \subseteq \delta \uparrow (\alpha, \beta) \\
\Longrightarrow & \ min(1 - \delta(x, y), \mu_{(\alpha,\beta)}(y)) \leq \alpha, \\
& max(\delta(x, y), \nu_{(\alpha,\beta)}(y)) \geq \beta \\
\Longrightarrow & \ \delta \uparrow (\alpha, \beta)(x) = (sup_{y \in U} min(1 - \delta(x, y), \mu_{(\alpha,\beta)}(y)), inf_{y \in U} max(\delta(x, y), \mu_{(\alpha,\beta)}(y))) \\
& \leq (\alpha, \beta)(x) \\
\Longrightarrow & \ \delta \uparrow (\alpha, \beta) \subseteq (\alpha, \beta) \\
\Longrightarrow & \ \delta \uparrow (\alpha, \beta) = (\alpha, \beta)
\end{aligned}
$$

□

Theorem 3.1.6. $(\delta \downarrow X^c)^c = \delta \uparrow X, \forall X \in IF(U), X^c$ is the complement of X

Proof.

$$
\begin{aligned}
& X^c = (\nu_X(x), \mu_X(x)), \forall x \in U \\
& \delta \downarrow X^c = (inf_{y \in U} max(\delta(x, y), \nu_X(y)), sup_{y \in U} min(1 - \delta(x, y), \mu_X(y))) \\
& (\delta \downarrow X^c)^c = (sup_{y \in U} min(1 - \delta(x, y), \mu_X(y)), \\
& inf_{y \in U} max(\delta(x, y), \nu_X(y))) = \delta \uparrow X(x)
\end{aligned}
$$

□

Theorem 3.1.7. $(\delta \uparrow X^c)^c = \delta \downarrow X, \forall X \in IF(U), X^c$ is the complement of X

Proof.

$X^c = (\nu_X(x), \mu_X(x)), \forall x \in U$

$\delta \uparrow X^c = (sup_{y \in U} min(1 - \delta(x, y), \nu_X(y)), inf_{y \in U} max(\delta(x, y), \mu_X(y)))$

$(\delta \uparrow X^c)^c = (inf_{y \in U} max(\delta(x, y), \mu_X(y)), sup_{y \in U} min(1 - \delta(x, y), \nu_X(y))) = \delta \downarrow X(x)$

□

Theorem 3.1.8. $\delta \downarrow (X \cap Y) = (\delta \downarrow X) \cap (\delta \downarrow Y)$

Proof.

$\delta \downarrow (X \cap Y)(x) = (inf_{y \in U} max(\delta(x,y), \mu_{X \cap Y}(y)), sup_{y \in U} min(1 - \delta(x,y), \nu_{X \cap Y}(y)))$

$= (inf_{y \in U} max(\delta(x,y), min(\mu_X(y), \mu_Y(y)), sup_{y \in U} min(1 - \delta(x,y), max(\nu_X(y), \nu_Y(y)))))$

$= (inf_{y \in U} min(max(\delta(x,y), \mu_X(y)), max(\delta(x,y), \mu_Y(y))),$

$sup_{y \in U} max(min(1 - \delta(x,y), \nu_X(y)), min(1 - \delta(x,y), \nu_Y(y))))$

$= (min(inf_{y \in U}(max(\delta(x,y), \mu_X(y)), inf_{y \in U}(max(\delta(x,y), \mu_Y(y))))),$

$max(sup_{y \in U}(min(1 - \delta(x,y), \nu_X(y))), sup_{y \in U}(min(1 - \delta(x,y), \nu_Y(y)))))$

$= \delta \downarrow X(x) \cap \delta \downarrow Y(x)$

□

Theorem 3.1.9. $\delta \uparrow (X \cap Y) \subseteq (\delta \uparrow X) \cap (\delta \uparrow Y)$

Proof.

$\delta \uparrow (X \cap Y)(x) = (sup_{y \in U} min(1 - \delta(x,y), \mu_{X \cap Y}(y)), inf_{y \in U} max(\delta(x,y), \nu_{X \cap Y}(y)))$

$= (sup_{y \in U} min(1 - \delta(x,y), min(\mu_X(y), \mu_Y(y))), inf_{y \in U} max(\delta(x,y), max(\nu_X(y), \nu_Y(y))))$

$= (sup_{y \in U} min(min(1 - \delta(x,y), \mu_X(y)), min(1 - \delta(x,y), \mu_Y(y))),$

$inf_{y \in U} max(max(\delta(x,y), \nu_X(y)), max(\delta(x,y), \nu_Y(y))))$

$\leq (min(sup_{y \in U}(min(1 - \delta(x,y), \mu_X(y))), sup_{y \in U}(min(1 - \delta(x,y), \mu_Y(y)))),$

$max(inf_{y \in U}(max(\delta(x,y), \nu_X(y)), inf_{y \in U}(max(\delta(x,y), \nu_Y(y))))))$

$= \delta \uparrow X(x) \cap \delta \uparrow Y(x)$

□

Theorem 3.1.10. $\delta \downarrow (X \cup Y) \supseteq (\delta \downarrow X) \cup (\delta \downarrow Y)$

Proof.

$\delta \downarrow (X \cup Y)(x) = (inf_{y \in U} max(\delta(x,y), \mu_{X\cup Y}(y)), sup_{y\in U} min(1-\delta(x,y), \nu_{X\cup Y}(y)))$

$= (inf_{y\in U} max(\delta(x,y), max(\mu_X(y), \mu_Y(y)), sup_{y\in U} min(1-\delta(x,y), min(\nu_X(y), \nu_Y(y)))))$

$= (inf_{y\in U} max(max(\delta(x,y), \mu_X(y)), max(\delta(x,y), \mu_Y(y))),$

$sup_{y\in U} min(min(1-\delta(x,y), \nu_X(y)), min(1-\delta(x,y), \nu_Y(y))))$

$\geq (max(inf_{y\in U}(max(\delta(x,y), \mu_X(y)), inf_{y\in U}(max(\delta(x,y), \mu_Y(y))))),$

$min(sup_{y\in U}(min(1-\delta(x,y), \nu_X(y))), sup_{y\in U}(min(1-\delta(x,y), \nu_Y(y)))))$

$= \delta \downarrow X(x) \cup \delta \downarrow Y(x)$

□

Theorem 3.1.11. $\delta \uparrow (X \cup Y) = (\delta \uparrow X) \cup (\delta \uparrow Y)$

Proof.

$\delta \uparrow (X \cup Y)(x) = (sup_{y\in U} min(1-\delta(x,y), \mu_{X\cup Y}(y)), inf_{y\in U} max(\delta(x,y), \nu_{X\cup Y}(y)))$

$= (sup_{y\in U} min(1-\delta(x,y), max(\mu_X(y), \mu_Y(y))), inf_{y\in U} max(\delta(x,y), min(\nu_X(y), \nu_Y(y))))$

$= (sup_{y\in U} max(min(1-\delta(x,y), \mu_X(y)), min(1-\delta(x,y), \mu_Y(y))),$

$inf_{y\in U} min(max(\delta(x,y), \nu_X(y)), max(\delta(x,y), \nu_Y(y))))$

$= (max(sup_{y\in U}(min(1-\delta(x,y), \mu_X(y))), sup_{y\in U}(min(1-\delta(x,y), \mu_Y(y)))),$

$min(inf_{y\in U}(max(\delta(x,y), \nu_X(y)), inf_{y\in U}(max(\delta(x,y), \nu_Y(y))))))$

$= \delta \uparrow X(x) \cup \delta \uparrow Y(x)$

□

Theorem 3.1.12. $\delta \uparrow (X \cap (\alpha, \beta)) = (\delta \uparrow X) \cap (\alpha, \beta)$

Proof.

$\delta \uparrow (X \cap (\alpha,\beta))(x) = (sup_{y\in U} min(1-\delta(x,y), \mu_{X\cap(\alpha,\beta)}(y)), inf_{y\in U} max(\delta(x,y), \nu_{X\cap(\alpha,\beta)}(y)))$

$= (sup_{y\in U} min(1-\delta(x,y), min(\mu_X(y),\alpha)), inf_{y\in U} max(\delta(x,y), max(\nu_X(y),\beta)))$

$= (sup_{y\in U} min(min(1-\delta(x,y), \mu_X(y)),\alpha), inf_{y\in U} max(max(\delta(x,y), \nu_X(y)),\beta))$

$= (min(sup_{y\in U}(min(1-\delta(x,y), \mu_X(y))),\alpha), max(inf_{y\in U}(max(\delta(x,y), \nu_X(y))),\beta)$

$= \delta \uparrow X(x) \cap (\alpha,\beta)(x)$

□

Theorem 3.1.13. $\delta \downarrow (X \cup (\alpha,\beta)) = (\delta \downarrow X) \cup (\alpha,\beta)$

Proof.

$\delta \downarrow (X \cup (\alpha,\beta))(x) = (inf_{y\in U} max(\delta(x,y), \mu_{X\cup(\alpha,\beta)}(y)), sup_{y\in U} min(1-\delta(x,y), \nu_{X\cup(\alpha,\beta)}(y)))$

$= (inf_{y\in U} max(\delta(x,y), max(\mu_X(y),\alpha)), sup_{y\in U} min(1-\delta(x,y), min(\nu_X(y),\beta)))$

$= (inf_{y\in U} max(max(\delta(x,y), \mu_X(y)),\alpha), sup_{y\in U} min(min(1-\delta(x,y), \nu_X(y)),\beta))$

$= (max(inf_{y\in U}(max(\delta(x,y), \mu_X(y)),\alpha)), min(sup_{y\in U}(min(1-\delta(x,y), \nu_X(y))),\beta))$

$= \delta \downarrow X(x) \cup (\alpha,\beta)(x)$

□

Theorem 3.1.14. If δ_1 and δ_2 are two intuitionistic fuzzy divergencne measures such that $\delta_1(X,Y) \leq \delta_2(X,Y), \forall X,Y \in IF(U)$, then $\delta_1 \downarrow X \subseteq \delta_2 \downarrow Y$ and $\delta_1 \uparrow X \supseteq \delta_2 \uparrow Y$

Proof. Given that $\delta_1(X,Y) \leq \delta_2(X,Y), \forall X,Y \in IF(U)$

So,

$$\delta_1(x,y) \leq \delta_2(x,y), \forall x,y \in U$$

$$\implies max(\delta_1(x,y), \mu_X(y)) \leq max(\delta_2(x,y), \mu_X(y)), \forall y \in U$$

Also,

$$1 - \delta_1(x,y) \geq 1 - \delta_2(x,y)$$

$$\implies min(1 - \delta_1(x,y), \nu_X(y)) \geq min(1 - \delta_2(x,y), \nu_X(y))$$

By property of infemum and supremum,

$$inf_{y \in U} max(\delta_1(x,y), \mu_X(y)) \leq inf_{y \in U} max(\delta_2(x,y), \mu_X(y)), \forall y \in U$$

and

$$sup_{y \in U} min(1 - \delta_1(x,y), \nu_X(y)) \geq sup_{y \in U} min(1 - \delta_2(x,y), \nu_X(y))$$

$$\implies \delta_1 \downarrow X(x) \subseteq \delta_2 \downarrow X(x)$$

Similarly,

$$min(1 - \delta_1(x,y), \mu_X(y)) \geq min(1 - \delta_2(x,y), \mu_X(y)), \forall y \in U$$

and

$$max(\delta_1(x,y), \nu_X(y)) \leq max(\delta_2(x,y), \nu_X(y)), \forall y \in U$$

By property of infemum and supremum,

$$sup_{y \in U} min(1 - \delta_1(x,y), \mu_X(y)) \geq sup_{y \in U} min(1 - \delta_2(x,y), \mu_X(y))$$

and

$$inf_{y \in U} max(\delta_1(x,y), \nu_X(y)) \leq inf_{y \in U} max(\delta_2(x,y), \nu_X(y))$$

$$\implies \delta_1 \uparrow X(x) \supseteq \delta_2 \uparrow X(x)$$

□

3.1.1 Feature selection using IF rough set model based on divergence measure

After establishing the various properties satisfied by lower and upper approximation. The equivalence relation produced by decision class is $U \setminus D = \{D_1, D_2, \ldots D_k\}$ with each decision class representing an intutiionisic fuzzy set given by:

$$D_i(x) = \begin{cases} (1,0), & if x \in D_i \\ \\ (0,1), & if x \notin D_i \end{cases}$$

Real valued conditional attribute A_i are converted to intuitionistic fuzzy set as:

$$\mu_{A_i}(x) = \frac{x-a}{b-a}, where\, a = min_{x\in U} A_i(x)\, and\, b = max_{x \in U} A_i(x)$$
$$\nu_{A_i}(x) = \frac{1-\mu_{A_i}(x)}{1+\lambda\mu_{A_i}(x)}$$

where non membership values are generated using Sugeno's intuitionistic fuzzy generator [18, 112]. The divergence Div_B is $n \times n$ matrix for $B \subseteq C$ where entries are given by $Div_B = [\delta_B(x_i, x_j)]_{n\times n}$, with $\delta_B(x_i,x_j) = max_{A\in B}\delta_A(x_i,x_j), i, j = 1, 2, \ldots, n$. Lower and upper approximations are calculated using the divergence matrix, which is employed to find positive region as:

$$Pos_A(x) = (sup_i \mu_{\delta\downarrow_{D_i} X}(x), inf_i \nu_{\delta\downarrow_A D_i}(x))$$

The degree of dependency is thereby given by $\gamma_A(Div) = \frac{\sum_{x\in U}|Pos_A(x)|}{|U|}$. The value of significance of $A \subseteq C$ with respect to $B \subseteq C$ such that $A \subseteq C - B$ is calculated using following formula:

$$Sig_A^B = \gamma_{A\cup B}(Div) - \gamma_A(Div)$$

The algorithm for feature selection is given as follows:

Algorithm 3.1.1 Divergence based feature selection

```
Input intuitionistic fuzzy information system
Find the decision class $U \setminus D = \{D_1, D_2, \ldots, D_k\}$
Initialize $C = \{a_1, a_2, \ldots, a_m\}$ and $Red = \phi$
repeat
    For every $a_i \in C - Red$, compute divergence matrix with respect to $Red \cup \{a_i\}$
    For $i = 1, 2, \ldots, m$, compute lower approximation $\delta \downarrow (Red \cup \{a_i\})(x) = (inf_{x\in U} max(\delta(x,y), \mu_{Red\cup\{a_i\}}), sup_{x\in U} min(1-\delta(x,y), \nu_{Red\cup\{a_i\}}))$, for each $x \in U$
    Compute positive region $Pos_{Red\cup\{a_i\}}(x) = (sup_j \mu_{\delta\downarrow Red\cup\{a_i\}(D_j)}(x),$
```

Table 3.1: Example dataset

Features Instances	a_1	a_2	a_3	a_4	D
x_1	(0, 1)	(0.33, 0.33)	(0.25, 0.42)	(0, 1)	1
x_2	(1, 0)	(0.66, 0.11)	(0, 1)	(0.33, 0.33)	1
x_3	(0, 1)	(0, 1)	(0, 1)	(0.66, 0.11)	2
x_4	(0, 1)	(1, 0)	(0.50, 0.20)	(0.33, 0.33)	2
x_5	(0.75, 0.7)	(0, 1)	(0.37, 0.29)	(1, 0)	1
x_6	(0.25, 0.47)	(0.33, 0.33)	(0.62, 0.13)	(0.66, 0.11)	1

Table 3.2: Divergence matrix obtained from example dataset for attribute a_1

	x_1	x_2	x_3	x_4	x_5	x_6
x_1	0	1	0	0	0.83	0.41
x_2	1	0	1	1	0.16	0.58
x_3	0	1	0	0	0.83	0.41
x_4	0	1	0	0	0.83	0.41
x_5	0.83	0.16	0.83	0.83	0	0.42
x_6	0.41	0.58	0.41	0.41	0.42	0

```
    $inf_j \nu_{\delta\downarrow Red\cup\{a_i\}(D_j)}(x)), \forall x \in U$
    Compute degree of dependency $\gamma_{Red\cup\{a_i\}}(Div) = \frac{\sum_{x\in U}|Pos_{Red\cup\{a_i\}}(x)|}{|U|}, \forall a_i \in C - Red$
    Compute significance $Sig_{a_i}^{Red\cup\{a_i\}}$
    Find attribute $a_t$ with greatest $Sig_{a_i}^{Red\cup\{a_i\}}$
until $Sig_{a_i}^{Red\cup\{a_i\}} > \epsilon', Red = Red \cup \{a_t\}$ and $C = C - Red$
return Red
```

The approach can be illustrated by a toy example given in table 3.1. The attribute 'D' is a decision attribute. Intuitionistic fuzzy divergence between x_i and x_j for $A \subseteq C$ is obtained by using formula: $\delta(x_i, x_j) = sup_{a_i \in A} \frac{(|\mu_{a_i}(x_i) - \mu_{a_i}(x_j)| + |\nu_{a_i}(x_i) - \nu_{a_i}(x_j)|}{2}$. Thereby, the divergence matrix for attribute a_1 is given in table 3.2. The decision class is partitioned as $U \setminus D = \{D_1, D_2\} = \{\{x_1, x_2, x_5, x_6\}, \{x_3, x_4\}\}$. Hence, the lower approimation is obained as in table 3.3. Degree of dependency of decision

Table 3.3: Lower approximation obtained from example dataset

	x_1	x_2	x_3	x_4	x_5	x_6
$U \setminus D_1$	(0, 1)	(1, 0)	(0, 1)	(0, 1)	(0.83, 0.16)	(0.41, 0.58)
$U \setminus D_2$	(0, 1)	(0, 1)	(0, 1)	(0, 1)	(0, 1)	(0, 1)

attribute over conditional attribute a_1 are calculated as $\gamma_{a_1} = \frac{2.2470}{6} = 0.3745$. Similarly, degree of dependencies of other attributes are computed as:

$$\gamma_{a_2} = \frac{1.4442}{6} = 0.2407$$
$$\gamma_{a_3} = \frac{1.4460}{6} = 0.2410$$
$$\gamma_{a_4} = \frac{0.7224}{6} = 0.1204$$

Hence, the significance is:

$$Sig_{a_1} = 0.3745$$

$$Sig_{a_2} = 0.2407$$

$$Sig_{a_3} = 0.2410$$

$$Sig_{a_4} = 0.1204$$

Since, attribute a_1 has highest significance value. So, $\{a_1\}$ is selected as potential reduct set. Similarly, the degree of dependencies is calculated by adding other attributes to this potential reduct set. The proposed algorithm iterates, and after termination of this algorithm, we obtain the reduct set as $\{a_1, a_2\}$.

3.1.2 Experimentation

The performance of proposed feature selection approach is evaluated on few benchmark datasets from UCI repository [108] as recorded in table 3.4 and 3.5, along with comparative analysis with Tan et. al. [140] and Neumann et. al. [114] approaches which clearly illustrates that the overall performance is usually enhanced for all the

Table 3.4: Benchmark datasets characteristics and reduct size

Dataset	Instances	Attributes	Reduct size		
			Tan et. al. approach	Neumann et. al. approach	Proposed algorithm
Balloon Scale+Stretch	20	4	2	2	2
Acute-inflammations-nephritis	120	7	2	3	3
Hill-valley-without-noise	1212	100	9	100	3
Appendicitis	106	7	5	7	3
Trains	10	26	2	23	2
Wpbc	194	33	10	na[2]	9

Table 3.5: Comparison of classification accuracies for original datasets and reduced datasets by proposed model, Tan et. al. and Neumann et. al. approach using 10-fold cross validation

Dataset	Original			Tan et. al. approach			Neumann et. al. approach			Proposed model		
	SVM	PART	Random forest	SVM	PART	Random forest	SVM	PART	Random forest	SVM	PART	Random forest
Balloon Scale+Stretch	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00
Acute-inflammations-nephritis	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00	97.58 ± 4.32	100.0 ± 0.00	100.0 ± 0.00	95.00 ± 6.04	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00
Hill-valley-without-noise	52.45 ± 3.23	50.49 ± 0.16	61.14 ± 4.06	49.73 ± 3.50	50.49 ± 0.16	54.55 ± 4.22	52.45 ± 3.23	50.49 ± 0.16	61.14 ± 4.06	49.56 ± 3.45	50.49 ± 0.16	53.72 ± 4.26
Appendicitis	86.95 ± 7.32	84.57 ± 9.35	86.98 ± 8.15	85.05 ± 9.01	82.16 ± 8.48	86.34 ± 9.35	86.95 ± 7.32	84.57 ± 9.35	86.98 ± 8.15	85.81 ± 8.49	84.55 ± 8.27	84.63 ± 8.81
Trains	0.00 ± 0.00	90.00 ± 30.15	50.00 ± 50.25	80.00 ± 40.20	90.00 ± 30.15	65.00 ± 47.94	90.00 ± 30.15	80.00 ± 40.20	69.00 ± 46.48	100.0 ± 0.00	90.00 ± 30.15	90.00 ± 30.15
Wpbc	44.87 ± 2.59	27.30 ± 9.36	40.96 ± 5.56	44.87 ± 2.59	30.54 ± 9.73	41.37 ± 5.42	na[3]	na[4]	na[4]	44.87 ± 2.59	29.50 ± 9.16	42.45 ± 6.30

reduced datasets. In order to avoid overfitting and unbiasedness, experiments are conducted based on 10-fold cross validation. Three classifiers namely SVM[1] [117], PART [49] and random forest [16] are used for experimentation. Different values of threshold parameter ϵ' were used for various datasets. Reduct produced by our approach is either smaller or similar than that produced by Tan et. al. [140] and Neumann et. al. [114] for all the datasets. Further, there was no reduction in case of appendicitis, and hill-valley without-noise using Neumann et. al. approach. However, the overall accuracies along with standard deviations of different learning algorithms for reduced datasets as produced by the proposed approach are better except for hill-valley-without-noise, wpbc, and appendicitis, where the performances of different classifiers are similar. Hence, the proposed approach outperforms the existing approaches.

3.1.3 Application to tuberculosis treatment

Tuberculosis, caused by Mycobacterium tuberculosis (M. tuberculosis), is a worldwide health malady that claims almost 1.8 million lives annually. In 'WHO Global

[1] SVM and SMO are used interchangeably in the study

[4] Neumann et. al. approach is not applicable on non-binary class dataset

Tuberculosis Report-2017', it was declared that Tuberculosis (TB) is one of the top ten leading causes of death world-wide [52]. In 2016, six lakhs new cases with resistance to rifampicin (RIF) having most effective first-line drug were reported, of which 490,000 cases was related with multidrug-resistant tuberculosis. This report declared TB as the leading cause of the deaths including HIV-associated TB deaths also. In 2015, 10.4 million cases were registered, of which 12% involved human immune deficiency virus (HIV) co-infection and 10% were among children. In 2015, 1.8 million deaths were reported due to tuberculosis including HIV-associated tuberculosis deaths also. It declared tuberculosis as the leading cause of death from an infectious disease [53]. Latent Mycobacterium tuberculosis infection is known as the reservoir of the TB epidemic. M. tuberculosis infection has newly been re-estimated at 24% of the global burden [61].The global decline rate of TB incidence is currently 1.5% and will be required to increase to 4%–5% and then to 10% per year by 2020 and 2025 respectively to achieve the World Health Organization End TB Strategy targets [6]. Mycobacterium consists distinguish characteristic [12, 136, 150], which creates an obstacle for the prediction of universal antibacterial peptides (ABP). Therefore, Usmani et. al. [145] presented an approach based on machine learning techniques for discriminating anti-tubercular with ABP as well as non-ABP. However, there are various aspects that can affect in obtaining the actual performance of the machine learning algorithms. The key factors among these are extraction of appropriate sequence features, selection of relevant and non-redundant features, and selection of suitable learning algorithms. In the current study, different sequence features of the peptides are extracted from Usmani et. al. dataset [145] to develop prediction models. In this dataset, anti-tubercular peptides (AntiTbP) were extracted from AntiTbPdb [146]. Final positive data consists of 246 unique peptides, varying in length from 5 to 61, which is effective against Mycobacterium. Final negative dataset consists of two separate datasets namely:

1. AntiTb_MD

2. AntiTb_RD

From DBAASP, peptides containing natural amino acids without any modifications were selected. These peptides represent active against Gram positive and Gram negative bacteria. One of the negative datasets containing 246 antibacterial peptides was selected from DBAASP. Moreover, 246 random peptides were created from Swiss-Prot, which was used for generating non-ABP dataset by eliminating AntiTbP and ABP. The ranges of peptide length are same in all three datasets. By extracting various features from positive and negative datasets, primary datasets consisting of 492 instances and 953 features and secondary datasets consisting of 492 instances and 953 features are created. Features are extracted by using iFeature web server [24]. These features are namely: amino acid composition, dipeptide composition, binary, moran correlation, composition/transition/distribution, pseudo amino acid composition, conjoint triad, quasi-sequence order, amino acid index, grouped dipeptide composition, and grouped tripeptide composition. By extracting appropriate features from the dataset, all the peptides are changed into numerical component vectors with a similar dimension. But not all the extracted features could contribute to classification accuracy. The high dimension feature vector may result in over-fitting, information redundancy, and dimensionality problem [36]. Feature Selection (FS), a powerful system to remove irrelevant and redundant features, is very valuable in decreasing the dimensionality of data and improving the classification accuracy. The schematic architecture of the proposed methodology is summarized in figure 3.1.

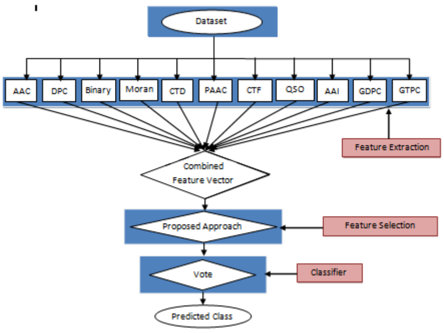


FIGURE 3.1: Flowchart of proposed methodology for prediction of anti tubercular peptides

3.1.3.1 Result

In this section, eight machine learning algorithms namely (1) support vector machine (SVM) [117] with puk kernel, (2) realAdaBoost with random forest classifier [15], also known as boosted random forest, (3) rotation forest [125], (4) PART [49], (5) J48 [120], (6) vote-based classifier [2] (with suitable classifiers combination), (7) random tree [79], and (8) random forest [16] are used. Among these eight classifiers, vote-based classifier produces highest accuracy of 87.80% for primary and 92.90% for secondary datasets. The proposed approach reduced the size of feature sets to 123 and 114 for extracted features of primary and secondary datasets respectively, but the performances of machine learning algorithms are better (or sometimes similar) for all the performance parameters when compared with the results for raw datasets, as recorded in table 3.6, 3.7, 3.8, 3.9, 3.10, 3.11, and 3.12. These results clearly illustrate that the approach reduces size of the dataset considerably without affecting the real performance of learning algorithms. All the experiments are performed using percentage split of 80:20 and 10 fold cross validation.

A comparative study of the proposed approach with model presented by Usmani et. al. is shown in table 3.15. The approach gave 87.80% and 92.90% average accuracy, 0.922 and 0.914 AUC which is much higher than 75.87% and 78.54% overall accuracy, 0.830 and 0.860 AUC as given by Usmani et. al. for primary and secondary datasets respectively. Further, table 3.13 and table 3.14 demonstrates the accuracy and AUC of various classifiers for reduced primary and secondary datasets obtained by using approach of Tan et. al. [140] and Neumann et. al. [113]. Model proposed by Tan et. al. is producing very poor performance as very few features are selected that might lead to information loss. Furthermore, an ensemble feature selection technique (presented by Neumann et. al.) is applied on both primary and secondary datasets. This approach eliminates only 11 and 20 features, which reduces the size of feature sets to 941 and 932 for primary and secondary datasets respectively. On comparing the results, it is obvious that the proposed approach is producing better results for both primary and secondary datasets. This clearly shows that our approach gives better performance and outperforms previous works. Receiver Operator Characteristics (ROC) is a convenient way to visualize the performance of different machine learning algorithms. Figure 3.2 is a plot of ROC for raw primary and secondary dataset, while ROC for reduced dataset is shown in figure 3.3. Moreover, the proposed approach was applied on various datasets from AMP databases consisting of anti-cancer peptides [167], antiviral peptides [141], anti-hypertensive peptides [106], and bitter peptides [21]. From the experimental results (table 3.16, 3.17, 3.18, 3.19 and 3.20), it can be concluded that the proposed method based results are always better than previously reported results. In case of anti-cancer, antiviral, anti-hypertensive, and bitter peptides prediction, Yi et. al. [167], Thakur et. al. [141], Manavalan et. al. [106], Charoenkwan et. al. [21] achieved the highest accuracy and AUC of 81.48 and 0.894, 85.70 and na (not available), 88.30 and 0.951, 84.38 and 0.904 respectively, while the proposed approach gives the best results as: 85.80 and 0.892, 86.40 and 0.945, 89.70 and 0.956, 86.7 and 0.929 respectively (table 3.20). These experimental results are clearly indicating that the approach is generalizable.

Table 3.6: Dataset characteristics and reduct size

Dataset	Number of Instances	Number of Attributes	Reduct size
Primary	492	953	123
Secondary	492	953	114

Table 3.7: Performance evaluation parameters of learning algorithms with raw primary dataset using percentage split of 80:20

Learning Algorithm	Sensitivity	Specificity	Accuracy	MCC	AUC	g-means
SVM	93.8	50.0	71.4	0.484	0.719	68.5
RealAdaBoost	85.4	74.0	79.6	0.597	0.885	79.5
RotationForest	68.8	78.0	73.5	0.470	0.835	73.3
PART	60.4	84.0	72.4	0.458	0.684	71.2
J48	64.6	80.0	72.4	0.452	0.711	71.9
Vote	85.4	74.0	79.6	0.597	0.892	79.4
RandomTree	72.9	52.0	62.2	0.254	0.625	61.6
RandomForest	95.8	72.0	83.7	0.696	0.887	83.1

Table 3.8: Performance evaluation parameters of learning algorithms with raw secondary dataset using percentage split of 80:20

Learning Algorithm	Sensitivity	Specificity	Accuracy	MCC	AUC	g-means
SVM	98.1	61.4	81.6	0.655	0.798	77.6
RealAdaBoost	96.3	79.5	88.8	0.778	0.966	87.5
RotationForest	87.0	79.5	83.7	0.669	0.924	83.1
PART	72.2	77.3	74.5	0.492	0.785	74.7
J48	68.5	70.5	69.4	0.338	0.694	69.5
Vote	94.4	77.3	86.7	0.736	0.969	85.4
RandomTree	68.5	86.4	76.5	0.550	0.774	76.9
RandomForest	90.7	84.1	87.8	0.752	0.944	87.3

In order to avoid overfitting and unbiasedness of our technique, experiments are conducted based on 10-fold cross validation also for tuberculosis dataset. The results are very promising and recorded in table 3.11 and 3.12. The best results obtained had overall sensitivity of 90.29, specificity of 84.25, accuracy of 87.16%, Mathews correlation coefficient (MCC) of 0.745, AUC of 0.928, which is much better when compared to the results presented by Usmani et. al. [145].

Table 3.9: Performance evaluation parameters of learning algorithms with reduced primary dataset using percentage split of 80:20

Learning Algorithm	**Sensitivity**	**Specificity**	**Accuracy**	**MCC**	**AUC**	**g-means**
SVM	91.5	71.2	80.8	0.635	0.813	80.7
RealAdaBoost	87.2	82.7	84.8	0.698	0.895	84.9
RotationForest	74.5	80.8	77.8	0.554	0.859	77.6
PART	72.3	75.0	73.7	0.473	0.743	73.6
J48	80.9	65.4	72.7	0.466	0.720	72.7
Vote	92.0	83.3	87.8	0.757	0.922	87.5
RandomTree	70.2	73.1	71.7	0.433	0.716	71.6
RandomForest	87.2	82.7	84.8	0.698	0.900	84.9

Table 3.10: Performance evaluation parameters of learning algorithms with reduced secondary dataset using percentage split of 80:20

Learning Algorithm	**Sensitivity**	**Specificity**	**Accuracy**	**MCC**	**AUC**	**g-means**
SVM	94.6	76.7	86.9	0.736	0.857	85.1
RealAdaBoost	89.3	88.4	88.9	0.775	0.930	88.8
RotationForest	80.4	86.0	82.8	0.659	0.879	83.1
PART	83.9	76.7	80.8	0.608	0.817	80.2
J48	82,1	74.4	78.8	0.567	0.776	78.1
Vote	96.4	88.4	92.9	0.857	0.914	92.3
RandomTree	73.2	83.7	77.8	0.565	0.785	78.2
RandomForest	89.3	90.7	89.9	0.796	0.934	89.9

Table 3.11: Performance evaluation parameters of learning algorithms with reduced primary dataset using 10-fold cross validation

Learning Algorithm	**Sensitivity**	**Specificity**	**Accuracy**	**AUC**	**MCC**	**g-means**
SVM	82.56	64.93	73.48	0.737	0.478	73.21
RealAdaBoost	78.34	75.38	76.54	0.847	0.534	76.84
RotationForest	70.06	71.86	70.81	0.789	0.418	70.95
PART	71.7	67.38	68.97	0.700	0.391	69.51
J48	64.08	66.04	65.09	0.674	0.300	65.05
Vote	77.42	75.38	76.53	0.843	0.535	76.39
RandomTree	65.62	73.72	69.59	0.697	0.393	69.55
RandomForest	78.87	73.46	75.93	0.838	0.521	76.11

Table 3.12: Performance evaluation parameters of learning algorithms with reduced secondary dataset using 10-fold cross validation

Learning Algorithm	**Sensitivity**	**Specificity**	**Accuracy**	**AUC**	**MCC**	**g-means**
SVM	90.84	74.62	82.26	0.827	0.660	82.33
RealAdaBoost	89.18	85.09	86.93	0.933	0.740	87.11
RotationForest	85.27	82.25	83.69	0.911	0.673	83.74
PART	76.68	78.68	77.97	0.809	0.558	77.67
J48	76.02	82.08	78.99	0.818	0.585	78.99
Vote	89.18	84.71	86.72	0.931	0.737	86.92
RandomTree	79.90	75.24	78.56	0.776	0.555	77.53
RandomForest	90.29	84.25	87.16	0.928	0.745	87.21

Table 3.13: Comparison of performance evaluation parameters of learning algorithms with reduced training set of primary dataset with percentage split of 80:20 using feature selection approaches of Tan et. al. and Neumann et. al.

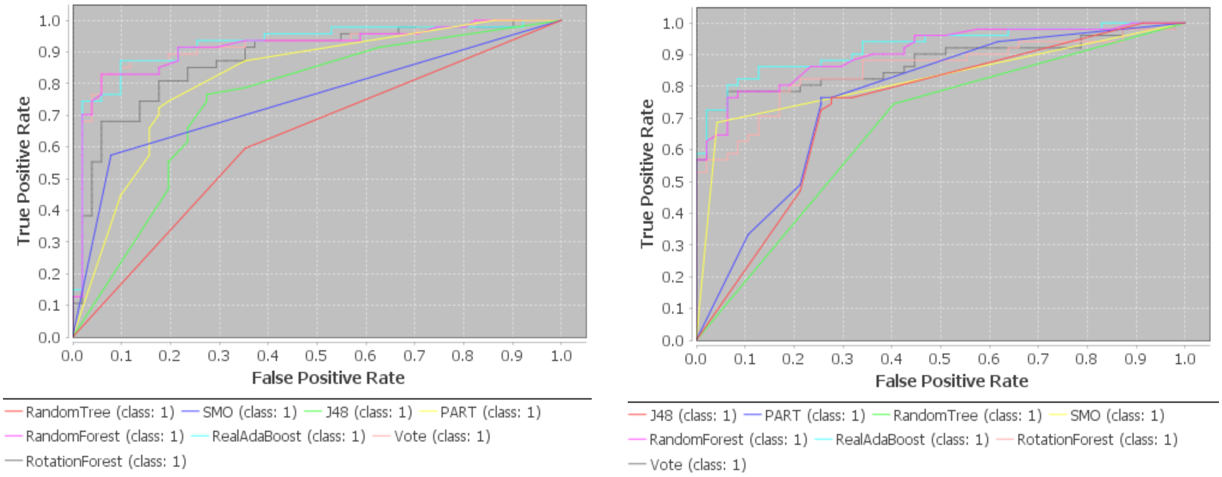
Learning Algorithm	**Tan et al Approach**		**Neumann et al Approach**		**Proposed Approach**	
	Accuracy	**AUC**	**Accuracy**	**AUC**	**Accuracy**	**AUC**
SVM	49.5	0.500	69.4	0.693	80.8	0.813
RealAdaBoost	49.5	0.500	65.3	0.701	84.8	0.895
RotationForest	49.5	0.500	73.5	0.811	77.8	0.859
PART	49.5	0.500	72.4	0.684	73.7	0.743
J48	49.5	0.500	72.4	0.711	72.7	0.720
Vote	49.5	0.500	81.6	0.877	87.8	0.922
RandomTree	49.5	0.500	63.3	0.631	71.7	0.716
RandomForest	49.5	0.500	77.6	0.872	84.8	0.900

Table 3.14: Comparison of performance evaluation parameters of learning algorithms with reduced training set of secondary dataset with percentage split of 80:20 using feature selection approaches of Tan et. al. and Neumann et. al.

Learning Algorithm	**Tan et al Approach**		**Neumann et al Approach**		**Proposed Approach**	
	Accuracy	**AUC**	**Accuracy**	**AUC**	**Accuracy**	**AUC**
SVM	44.5	0.500	83.7	0.837	86.9	0.857
RealAdaBoost	44.5	0.500	74.5	0.807	88.9	0.930
RotationForest	44.5	0.500	87.8	0.944	82.8	0.879
PART	44.5	0.500	74.5	0.785	80.8	0.817
J48	44.5	0.500	69.4	0.694	78.8	0.776
Vote	44.5	0.500	89.8	0.968	92.9	0.914
RandomTree	44.5	0.500	71.4	0.718	77.8	0.785
RandomForest	44.5	0.500	83.7	0.946	89.9	0.934

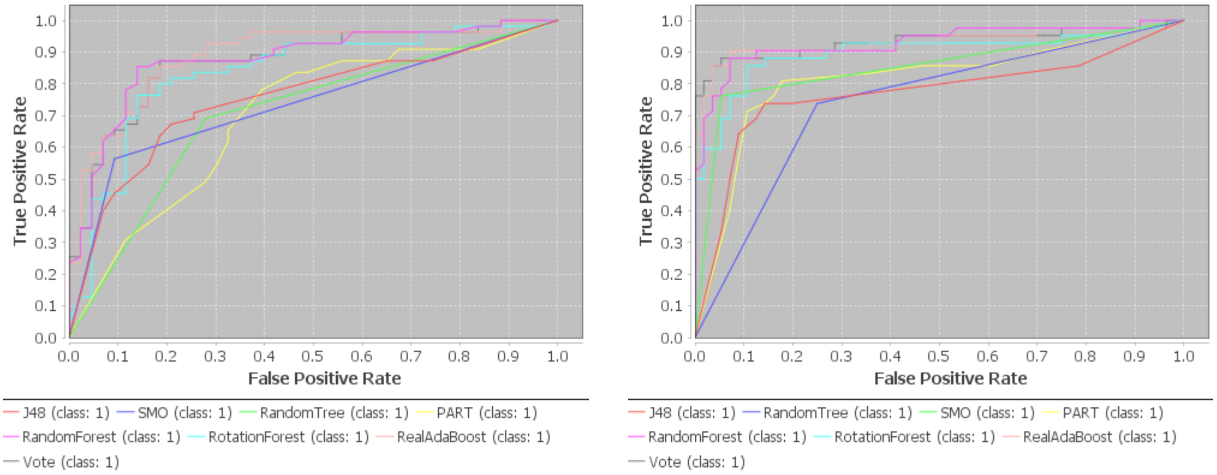
Table 3.15: Comparison of the performance evaluation metrics of the current work with the previous method

Method	Dataset	Accuracy	AUC
Usmani et. al.	**Primary**	75.87	0.830
	Secondary	78.54	0.860
Proposed method	**Primary**	87.80	0.922
	Secondary	92.90	0.914



Primary Secondary

Figure 3.2: AUC of eight machine learning algorithms for the dataset



Primary Secondary

Figure 3.3: AUC of eight machine learning algorithms for the reduced dataset

Table 3.16: Performance evaluation parameters of learning algorithms with reduced Yi et. al. dataset (ACP_740) using percentage split of 80:20

Learning Algorithm	**Sensitivity**	**Specificity**	**Accuracy**	**AUC**	**MCC**	**g-means**
SVM	80.2	77.6	79.1	0.789	0.578	78.89
RealAdaBoost	86.4	83.6	85.1	0.893	0.700	84.99
RotationForest	82..7	71.6	77.7	0.832	0.548	76.95
PART	77.8	67.2	73.0	0.742	0.542	72.30
J48	72.8	62.7	68.2	0.678	0.357	67.56
Vote	86.4	83.6	85.1	0.896	0.700	84.99
RandomTree	74.1	67.2	70.9	0.706	0.413	70.56
RandomForest	88.9	82.1	85.8	0.893	0.713	85.43

Table 3.17: Performance evaluation parameters of learning algorithms with reduced Thakur et. al. dataset (Antiviral) using percentage split of 80:20

Learning Algorithm	**Sensitivity**	**Specificity**	**Accuracy**	**AUC**	**MCC**	**g-means**
SVM	81.5	85.5	83.2	0.835	0.665	83.48
RealAdaBoost	88.0	84.3	86.4	0.944	0.723	86.13
RotationForest	88.0	77.1	83.2	0.919	0.658	82.37
PART	83.3	74.7	79.6	0.846	0.583	78.88
J48	85.2	74.7	80.6	0.836	0.604	79.78
Vote	88.0	84.3	86.4	0.945	0.723	86.13
RandomTree	77.8	69.9	74.3	0.738	0.477	73.74
RandomForest	87.0	85.5	86.4	0.944	0.724	86.25

Table 3.18: Performance evaluation parameters of learning algorithms with reduced Manavalan et. al. dataset (Anti-hypertensive) using percentage split of 80:20

Learning Algorithm	**Sensitivity**	**Specificity**	**Accuracy**	**AUC**	**MCC**	**g-means**
SVM	84.5	79.8	81.9	0.821	0.640	82.12
RealAdaBoost	90.1	86.9	88.4	0.958	0.768	88.49
RotationForest	88.7	82.1	85.2	0.935	0.706	85.34
PART	83.1	84.5	83.9	0.859	0.676	83.80
J48	83.1	79.8	81.3	0.860	0.627	81.43
Vote	90.1	86.9	88.4	0.956	0.768	88.49
RandomTree	87.3	79.8	83.2	0.835	0.669	83.47
RandomForest	91.5	88.1	89.7	0.956	0.744	89.78

Table 3.19: Performance evaluation parameters of learning algorithms with reduced Charoenkwan et. al. dataset (Bitter) using percentage split of 80:20

Learning Algorithm	**Sensitivity**	**Specificity**	**Accuracy**	**AUC**	**MCC**	**g-means**
SVM	87.5	78.1	82.8	0.828	0.659	82.67
RealAdaBoost	84.4	85.9	85.2	0.904	0.703	85.15
RotationForest	87.5	84.4	85.9	0.908	0.719	85.94
PART	71.9	85.9	78.9	0.836	0.584	78.59
J48	81.3	82.8	82.0	0.864	0.641	82.05
Vote	85..9	87.5	86.7	0.929	0.734	86.70
RandomTree	73.4	82.8	78.1	0.781	0.565	77.96
RandomForest	84.4	81.3	82.8	0.925	0.657	82.84

Table 3.20: Comparison of the performance evaluation metrics of the current work with the previous method

Methods	Dataset	Accuracy	AUC
Yi et. al.	**ACP_740**	81.48	0.894
Proposed method	**ACP_740**	85.80	0.892
Thakur et. al.	**Antiviral**	85.70	na
Proposed method	**Antiviral**	86.40	0.945
Manavalan et. al.	**Anti-hypertensive**	88.30	0.951
Proposed Method	**Anti-hypertensive**	89.70	0.956
Charoenkwan et. al.	**Bitter**	84.38	0.904
Proposed method	**Bitter**	86.7	0.929

3.2 k-mean based Intuitionistic Fuzzy Rough Set Model

An intuitionistic fuzzy rough set model based on k-mean is proposed in this work. The conventional intuitionistic fuzzy lower and upper approximations are prone to noise. The infimum and supremum are very sensitive to noisy sample objects and might lead to misleading results in presence of outliers. Since, a single noisy sample might affect value of infimum or supremum, this would further degrade the performance in case of noisy datasets. To overcome such situations, k nearest neighbour of an object is used to compute approximations. Hence, the lower and upper approximations are reformulated as:

$$R\downarrow_A D_i(x)=(\mu_{R\downarrow_A D_i}(x),\nu_{R\downarrow_A D_i}(x))=(\frac{1}{k}\sum_{l=1}^{k}{}_{y\notin D_i}\nu_{R_{A_l}}(x,y),\frac{1}{k}\sum_{l=n-k}^{n}{}_{y\notin D_i}\mu_{R_{A_l}}(x,y)) \tag{3.7}$$

$$R\uparrow_A D_i(x)=(\mu_{R\uparrow_A D_i}(x),\nu_{R\uparrow_A D_i}(x))=(\frac{1}{k}\sum_{l=n-k}^{n}{}_{y\in D_i}\mu_{R_{A_l}}(x,y),\frac{1}{k}\sum_{l=1}^{k}{}_{y\in D_i}\nu_{R_{A_l}}(x,y)) \tag{3.8}$$

where n samples of $\mu_{R_A}(x,y)$ $\{\mu_{R_{A_1}}(x,y),\mu_{R_{A_2}}(x,y),\ldots\mu_{R_{A_n}}(x,y)\}$ and $\nu_{R_A}(x,y)$ $\{\nu_{R_{A_1}}(x,y),\nu_{R_{A_2}}(x,y),\ldots\nu_{R_{A_n}}(x,y)\}$ are sorted in increasing order of their magnitudes.

Intuitionistic fuzzy positive region is defined using above formulated lower approximation and hence the degree of dependency.

Proposition 3.2.1. The k-means based intuitionistic fuzzy lower and upper approximations are intuitionistic fuzzy set in U.

Proof. Since, R_A is intuitionistic fuzzy set in $U \implies R\downarrow_A D_i$ is an intuitionistic fuzzy set in U. □

3.2.1 Feature Selection based on Intuitionistic Fuzzy Rough Set Model based on k-means

A greedy forward quick reduct algorithm for feature selection is employed in which dependency function is used as evaluation criterion. Hence, an attribute subset *Red* $\subseteq C$ is considered as potential reduct if its dependency does not increase on addition of attributes or its dependency is equal to dependency of whole feature set and decreases on removal of feature.

3.2.2 Application for Enhancing Prediction of Aptamer-protein Interacting Pairs

Aptamers, first discussed by [43],[144], are artificially synthesized single stranded DNA/RNA molecules or peptide molecules ([80]). These artificial molecules fold into specific three dimensional configurations and can combine to definite targets with exceptionally high specificity and affinity. Aptamers are found to be encouraging reagents in new drug development and protein detection due to their molecular recognition ability for proteins. In the area of therapeutics, biosensing and diagnostics, aptamers give good latent because of powerful ligands class. Consequently, aptamers–target interactions have drawn greater attention and have broad applications in the fundamental research. However, characterization of aptamers and experimental identification is usually costly and time consuming. Therefore, it is the essential requirement to establish a computational methodology for effectively and abruptly determining the aptamer-protein interacting pairs by taking into account the DNA, RNA and primary sequence information of protein.

Prediction of aptamer-target interacting pairs has been reported by only three computational methods. The first prediction model was presented by [89], which was based on random forest. [171] presented another approach by using an ensemble classifier containing three random forest sub-classifiers to overwhelm the imbalancing issue. [165] proposed third approach by using ensemble classifier method, which was a three SVMs (support vector machine) ensemble classifier.

There are various issues, which directly affect the learning of different classifiers. Extraction of suitable features, imbalance between classes, selection of non-redundant and relevant features, and selection of appropriate learning algorithms are considered as key issues in case of biological datasets. Here, dataset from [171] was used with the same extracted features. A few of the attributes may be irrelevant and/or redundant for classification task; they may extremely reduce the overall accuracy of classification learning and lead to a great degree of computational complexity. Therefore, before employing the dataset, it is needed to preprocess the data to eliminate redundant and non-predictive features.

3.2.2.1 Result

Seven different machine learning algorithms were employed to evaluate performance. Firstly, irrelevant and/or redundant features were removed by using the proposed k-mean approach. The value of k was set to 5. The number of attributes is reduced to 207 using k-mean based feature selection approach as shown in table 3.21. As the dataset is imbalanced, before evaluating performance synthethic minority oversampling technique (SMOTE) at 200% is applied to obtain optimal balancing ratio. Classification performance is then obtained using ten-fold cross validation. Among these seven classifiers, boosted random forest gave highest overall accuracy of 91.60% and 85.60% and sensitivity of 91.30% and 86.40% for training and testing set respectively. The values of the various performance evaluation metrics clearly demonstrate the effectiveness of the k-mean model. The values of various performance evaluation metrics for reduced training and test sets are given in table 3.22, 3.23. Also, classification performance along with standard deviation evaluated by paired t-test is also illustrated in table 3.24.

Further, a comparative study of the proposed k-mean approach with models of Li et. al. [89], Zhang et. al. [171] and Yang et. al. [165] is shown in Table 3.25, 3.26. This approach gave a maximum average accuracy of 91.60% for training and 85.60% for testing dataset respectively with improved values of other parameters, which is much higher than previous approaches. The values of sensitivity, MCC is higher using proposed kmean approach for both training and testing datasets. A small

Table 3.21: Dataset characteristics and reduct size

Dataset	Number of Instances	Number of Attributes	Reduct Size
Aptamer Train	2329	291	207
Aptamer Test	580	291	207

Table 3.22: Comparison of performance evaluation metrics for reduced training datasets by proposed model

Learning Algorithm	Sensitivity	Specificity	Accuracy	MCC	AUC	g-means
BayesNet	70.5	60.5	65.5	0.311	0.711	65.3
SVM	84.7	87.1	85.9	0.718	0.859	85.8
IBK	94.4	76.1	85.3	0.718	0.855	84.7
Rotation Forest	86.0	86.0	86.0	0.720	0.938	86.0
RealAdaBoost	91.3	91.9	91.6	0.832	0.969	91.5
PART	84.3	54.8	69.5	0.409	0.760	67.9
Random Forest	89.5	88.6	89.1	0.781	0.957	89.0

Table 3.23: Comparison of performance evaluation metrics for reduced testing datasets by proposed model

Learning Algorithm	Sensitivity	Specificity	Accuracy	MCC	AUC	g-means
BayesNet	65.1	53.6	59.3	0.187	0.633	59.0
SVM	76.6	77.5	77.0	0.540	0.770	77.0
IBK	89.4	75.2	82.3	0.653	0.823	81.9
Rotation Forest	81.1	77.5	79.3	0.587	0.863	79.2
RealAdaBoost	86.4	84.8	85.6	0.713	0.908	85.5
PART	66.2	64.8	65.5	0.310	0.673	65.4
Random Forest	84.1	83.0	83.6	0.671	0.881	83.5

decrease in specificity is observed but that is overruled by the significant increase in values of other paramters.

Performance of different machine learning algorithms can be visualised using Receiver Operating Characteristic (ROC) plot. ROC plot for reduced training and testing dataset is demonstrated in figure 3.4. The plot demonstrates that the plot corresponding to boosted random forest classifier is producing highest performance for both the training and testing datasets.

TABLE 3.24: Comparison of average classification accuracies along with standard deviation for reduced training and testing datasets using proposed approach

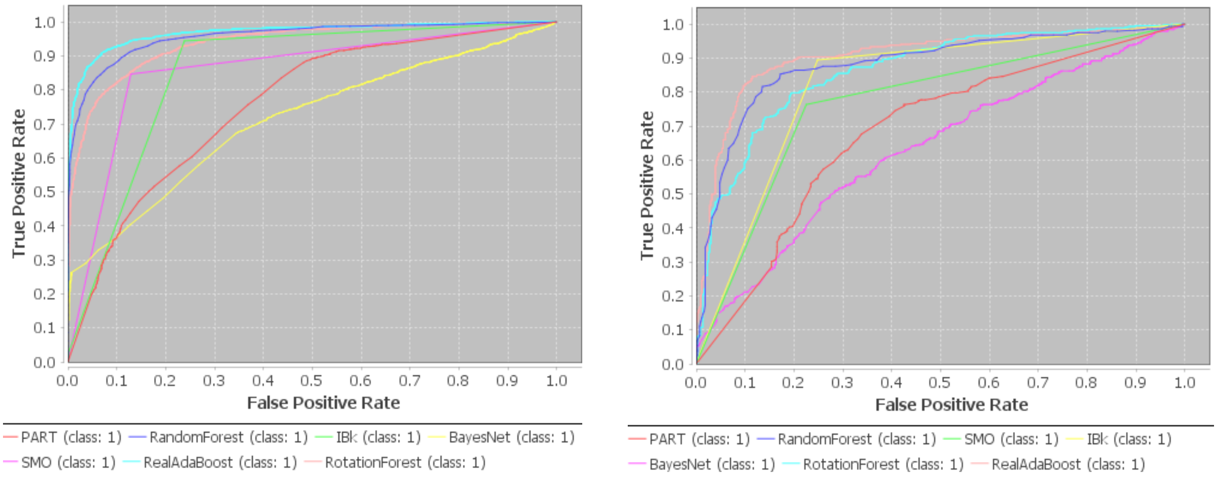
Learning Algorithm	**Training**	**Testing**
BayesNet	65.47 ± 2.41	60.18 ± 4.63
SVM	85.87 ± 1.85	76.85 ± 4.29
IBK	85.56 ± 1.80	81.32 ± 4.20
Rotation Forest	86.11 ± 1.87	78.86 ± 4.26
RealAdaBoost	91.94 ± 1.51	84.97 ± 3.81
PART	71.17 ± 3.93	67.37 ± 4.61
Random Forest	89.38 ± 1.72	82.71 ± 4.20

TABLE 3.25: Comparison of the best values of the performance evaluation metrics of the current work with the values of previous existing methods on training dataset

Learning Algorithm	**Sensitivity**	**Specificity**	**Accuracy**	**MCC**
Li et. al.	48.8	92.2	81.3	0.461
Zhang et. al.	75.3	72.5	73.2	0.424
Yang et. al.	77.3	73.7	74.5	0.450
Proposed method	91.3	91.9	91.6	0.832

TABLE 3.26: Comparison of the best values of the performance evaluation metrics of the current work with the values of previous existing methods on testing dataset

Learning Algorithm	**Sensitivity**	**Specificity**	**Accuracy**	**MCC**
Li et. al.	48.3	87.1	77.4	0.372
Zhang et. al.	73.8	71.3	71.9	0.398
Yang et. al.	79.3	74.5	75.7	0.478
Proposed method	86.4	84.8	85.6	0.713



Training Testing

Figure 3.4: AUC of seven machine learning algorithms for the reduced dataset

3.3 Summary

Intuitionistic fuzzy rough set based feature selection is presented in this chapter. Divergence based intuitionistic fuzzy rough set model is proposed along with the proofs of respective properties of lower and upper approximation. This model is employed for dimensionality reduction. Experimental and comparative study illustrate the utility of the proposed model. Further, it was applied to solve real life application to treat tuberculosis by prediction of anti-tubercular peptides. Likewise, a k-mean based intuitionistic fuzzy rough set model is introduced that is robust to noisy samples and applied for prediction of aptamer-protein interacting pairs. However, these concept can only maintain a maximal dependency function. It cannot preferably illustrate the differences in object classification and does not fit a particular data set well. This problem is handled in the upcoming chapter.

****