Inconsistent Decision System: Rough Set Data Mining Strategy to Extract Decision Algorithm of a Numerical Distance Relay – Tutorial

Mohammad Lutfi Othman and Ishak Aris

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/50460

1. Introduction

Modern numerical protective relays being intelligent electronic devices (IED) are inevitably vulnerable to false tripping or failure of operation for faults in the power system [1]. With regular and rigorous analyses the performance reliability of the digital protective relays can be ascertained, their availability maximized and subsequently their misoperation risks minimized [2]. The precise relay operation analyses would normally be assessing the relay characteristics, evaluating the relay performance and identifying the relay-power system interactions so as to ensure that the protective relays operate in correspond to their predetermined settings [3,4].

Protection engineers would in practice resort to computing technologies for automating the analysis process when the gravity of event data exploration, manipulation and inferencing incapacitate human manageability. The voluminous amount of data to be processed has prompted the need to use intelligent data mining, an essential constituent in the Knowledge Discovery in Databases (KDD) process [5]. This has motivated the adoption of rough set theory to data mine the protective relay event report so as to discover its decision algorithm.

2. Problem statement and objective

The following two pertinent problems are the attributing factors in driving this paper into studying the protective relay operation analysis:

- Inconsistencies in the device's event report particularly found when upon power system fault inception, a protective relay detects and invokes a common combination of tripping conditions in time succession but having two distinct tripping decisions (classifications). These distinct decisions are one, that upon relay pick-up, trip signal has not been asserted immediately after and the other is when a subsequent trip signal is asserted, after a preset time delay as set by the protection engineer.

- Non-linear nature of relay operation that makes it very difficult to select a group of effective attributes to fully represent relay tripping behavior.

In the grueling manual analysis of relay event report [1,6], the selected attributes hardly provide adequate knowledge in accurately mapping the interclass boundary in the relay decision system due to inconsistency. This characterizes the interclass boundary to be usually "rough". Based on the selected attributes, some relay events close to the boundary are unclassifiable – trip or nontrip. The small overlaps between different relay events make the protective relay operation analysis to be actually a rough classification problem. Thus, rough set theory has been appropriately chosen to resolve this conflict [7].

3. Rough set data mining in dealing with inconsistent numerical distance relay decision system to extract decision algorithm – The fundamental concept

Using rough set theory approach, relay decision rule extraction is naturally a byproduct of the data reduction process involved and easily understood. Rule extraction technique is inherent to the machine learning process of rough set theory. Thus, the inherent capability of rough set theory to discover fundamental patterns in relay data has essentially mooted this study. Using an approximation concept, rough set theory is able to remove data redundancies and consequently generate decision rules. In contrast to crisp sets, a rough set has boundary line cases – events that cannot be certainly classified either as members of the set or of its complement. Rough set theory is an alternative intelligent data analysis tool that can be employed to handle vagueness and inconsistencies [8].

An *information system* (*IS*) also alternatively known as *knowledge representation system* (*KRS*) is a tabulated data set, the rows of which are labeled by *objects* (events) of interest, columns labeled by *attributes*, and the entries are *attribute values* [8]. This data layout fits very well the protective relay event report that is characterized by its attributes of relay multifunctional elements versus sequence of time-stamped events [7].

In the protective relay event report, the *IS* manifestation is more appropriately referred as relay *decision table* or *decision system* (*DS*) as Huang et. al. [9] put it that decision table is characterized by disjoint sets of *condition attributes* ($C \subset Q$) and *decision (action) attributes* ($D \subset Q$). In this regard $Q = C \cup D$ and $C \cap D = \emptyset$. This *DS* is a 4-tuple structure formulated as $DS = \langle U, Q, V, f \rangle$, the elements of which are as follows [8,10,11]:

- U, i.e. the *universe* denoted as $U = \{t_1, t_2, t_3, \ldots, t_m\}$, is a finite set of relay events (t_i's).
- $Q = C \cup D$ is a non-empty finite union set of condition and decision *attributes*,
    - condition attributes ($c_i \subset C$) indicate the internally various multifunctional protective elements and analog measurands,

- decision attribute ($d_i \subset D$) indicates the trip output of the relay, such that q: $U \rightarrow V_q$ for every $q \in Q$.

- $V = U_{q \in Q} V_q$, where V_q is a set of values (*domain*) of the attribute q.
- f: $U \times Q \rightarrow V$ called *information function* is a total function such that $f(t,q) \in V_q$ for every $t \in U$, $q \in Q$. Any pair (q,v) is called *descriptor* in DS, where $q \in Q$ and $v \in V_q$,.

3.1. Relay decision system indiscernibility relation

If a set of attributes $P \subseteq Q = C \cup D$ and $f(t_x,q) = f(t_y,q)$ where t_x, $t_y \in U$, then for every $q \in P$, t_x and t_y are *indiscernible* (indistinguishable) by the set of attributes P in DS. Thus, every $P \subseteq Q$ brings forth a binary relation on U called *P-indiscernibility relation* (or *equivalence relation*) which is denoted by $IND(P)$. This suggests that there will be sets of relay events that are indiscernible based on any selected subset of attributes P. $U|IND(P)$ denotes the family of all *equivalence classes* of relation $IND(P)$. $IND(P)$ and $U|IND(P)$ can be formulated as

$$IND(P) = \{(t_x, t_y) \in U^2 | \forall q \in P, q(t_x) = q(t_y)\}, \tag{1}$$

$$U|IND(P) = \otimes\{q \in P | U|IND(\{q\})\}, \tag{2}$$

where,

$$A \otimes B = \{X \cap Y | \forall X \in A, \forall Y \in B, X \cap Y \neq \emptyset\}. \tag{3}$$

$U|IND(P)$ is also interchangeably referred as *P-basic knowledge* or *P-elementary sets* in DS. P-elementary set including relay event t is denoted as $[t]_{IND(P)}$. The first step in classification with rough sets is the construction of elementary sets [11]. A description of P-elementary set $X \in U|IND(P)$ in terms of values of attributes from P is denoted as $Des_P(X)$, i.e.

$$Des_P(X) = \{(q,v) : f(t,q) = v, \forall t \in X, \forall q \in P\} \tag{4}$$

3.2. Relay decision system set approximation

In the context of protective relay operations, consider $T \subseteq U$ as an arbitrary target set of relay events described (classified) by a particular trip assertion status that is needed to be represented by equivalence classes originating from attribute subset $P \subseteq Q$. P could be a selected condition attribute set $P \subseteq C$ or all condition attributes C reflecting relay multifunctional protective elements while T could be the set of relay events indiscernible with respect to the decision attribute $D = Trip$ having a domain value 'b' for pole-B tripping, for example [7].

The idea of the rough set revolves around the concept of approximation [11]. Thus, by introducing a pair of sets, called the *lower* and *upper approximations* of the target set T using only the information contained within P, the target set T can be *approximated*.

Formally, with a given relay decision system *DS*, each target subset $T \subseteq U$ having equivalence relation $IND(P)$ is related to two subsets of T as follows.

P-lower approximation of T expressed as,

$$\underline{P}T = \cup\{X \in U \big| IND(P) : X \subseteq T\}, \tag{5}$$

is defined as the union of all elementary sets in $[t]_{IND(P)}$ which are contained in T. For any relay event t_i of the lower approximation of T with respect to the set of attributes P (i.e., $t_i \in \underline{P}T$), it *positively certain* belongs to T.

P-upper approximation of T expressed as,

$$\overline{P}T = \cup\{X \in U \big| IND(P) : X \cap T \neq \varnothing\}, \tag{6}$$

is defined as the union of elementary sets in $[t]_{IND(P)}$ which have a non-empty intersection with T. For any relay event t_i of the upper approximation of T with respect to the set of attributes P (i.e., $t_i \in \overline{P}T$), it *may possibly* belong to T.

P-boundary of set T expressed as,

$$BN_P(T) \ = \overline{P}T - \underline{P}T \tag{7}$$

is the difference between $\overline{P}T$ and $\underline{P}T$. The set of elements t_i which *cannot be certainly* classified as belonging to T using the set of attributes P [12].

The following three regions shall be derived from the lower- and upper-approximations as illustrated in Figure 1 [7,10,13].

- $POS_P(T) = \underline{P}T$, described as P-positive region of T, is the set of relay events which can be classified with certainty in the approximated set T.
- $NEG_P(T) = U - \underline{P}T$, described as P-negative region of T, is the set of relay events which cannot be classified without ambiguity in the approximated set T (or classified as belonging to the complement of T).
- $BN_P(T) = \overline{P}T - \underline{P}T$, described as P-boundary region of T, is the set of relay events in which none can be classified with certainty into T nor its complement $\overline{T}$ as far as the attributes P are concerned. The set T is *crisp* if there are no boundary sets, i.e. $BN_P(T) = \varnothing$ (empty set), which otherwise it is *rough*.

3.3. Approximation accuracy and quality

$\alpha_P(T)$, the accuracy of the rough set representation of a target set of relay events T, is formulated as [10]

$$\alpha_P(T) = \frac{|\underline{P}T|}{|\overline{P}T|} = \frac{card(\underline{P}T)}{card(\overline{P}T)} . \tag{8}$$

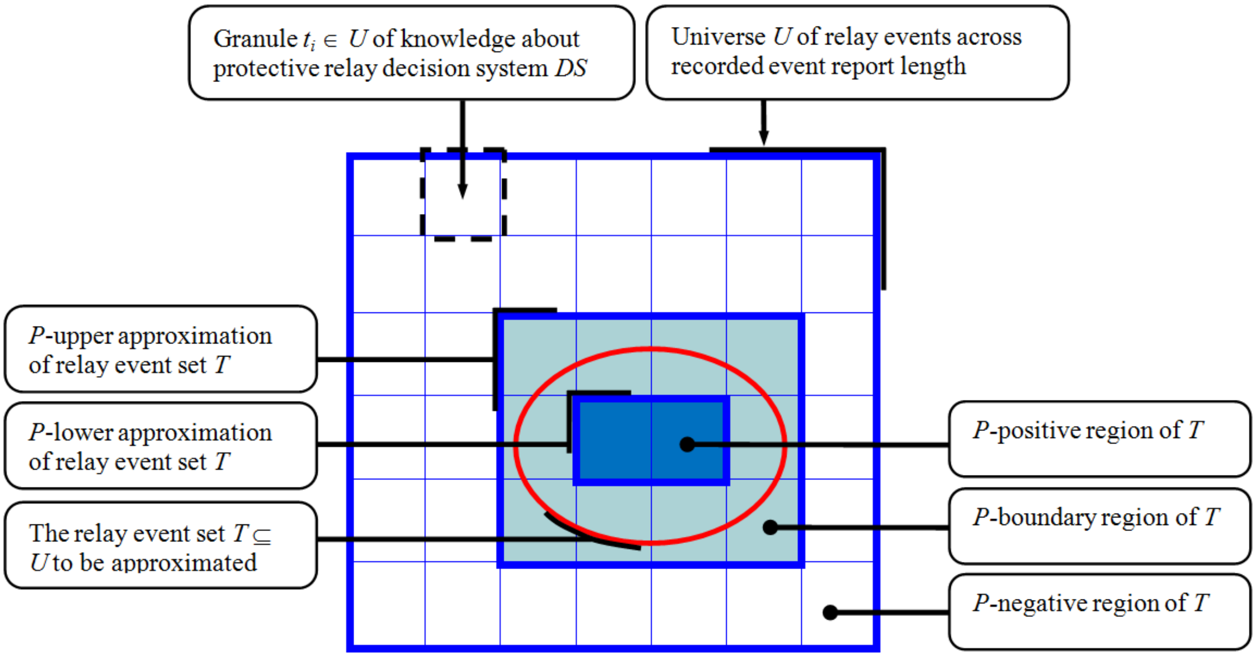

Figure 1. Definition of approximation in rough set theory in the context of protective relay

It provides a measure of how accurate the rough set is in approximating the target set of relay events T by comparing the number of events which can be *positively* placed in P with the number of events that can be *possibly* be placed in P. Noticeably $0 \le \alpha_P(T) \le 1$. (Note: *card* (cardinality) of a set is the number of events contained in the set [11]).

Clearly, equal upper and lower approximations, i.e. empty boundary region and that $\alpha_P(T) = 1$, would mean the target set T is said to be *definable* in U since it is perfectly approximated. Regardless of the size of the upper approximation, zero accuracy would mean the lower approximation is empty.

In general, the set T can be defined in U according to one of the following four concepts of definability [14,15]:

- *Roughly definable* T in U given $\underline{P}T \neq \varnothing$ and $\overline{P}T \neq U$ ($\varnothing$ denotes empty set)
- *Externally undefinable* T in U given $\underline{P}T \neq \varnothing$ and $\overline{P}T = U$
- *Internally undefinable* T in U given $\underline{P}T = \varnothing$ and $\overline{P}T \neq U$
- *Totally undefinable* T in U given $\underline{P}T = \varnothing$ and $\overline{P}T = U$

The *quality of approximation* of a target set T is expressed as

$$\gamma_P(T) = \frac{|\underline{P}T|}{U} = \frac{card(\underline{P}T)}{card(U)}, \tag{9}$$

i.e. the ratio of *P-correctly* approximated events to all events in the system.

3.4. The concept of reduct and core in reduction of protective relay attributes

Dependencies between attributes are primarily important in the protective relay data analysis using rough sets approach. The set of attributes $R \subseteq Q$ *depends* on the set of attributes $P \subseteq Q$ in *IS* if and only if $IND(P) \subseteq IND(R)$. This dependency is denoted as $P \rightarrow R$.

This so-called attribute reduction is so performed that the reduced set of attributes provides the same approximation quality as the original set of attributes. If a particular set of attributes is dependent, it is interesting to find *reducts* (all possible minimal subsets of attributes) that lead to the same number of elementary sets as in the case of the whole set of attributes and also to find *core* (the set of all indispensable attributes) [11]. By adopting the fundamental concepts of core and reduct, rough set theory minimizes the subsets of attributes in the relay database but still fully characterizes the inherent knowledge of relay operation behavior.

Reduct is essentially a sufficient set of features of a *DS*, which discerns (differentiates) all events discernible by the original *DS*. Reduct is a subset of attributes $RED \subseteq P$ (where $P \subseteq Q$) such that:

- The reduced attribute set *RED* induces the same equivalence classes as those induced by full attribute set *P*. This is denoted as $[t]_{IND(RED)} = [t]_{IND(P)}$.
- Attribute set *RED* is *minimal* in the sense that $[t]_{IND(RED-A)} \neq [t]_{IND(P)}$ for any attribute $A \in RED$. This suggests that no attribute can be dispensed from set *RED* without modifying the equivalence classes $[t]_{IND(P)}$ [16].

Core is defined as the set of attributes found to be in common in all reducts. Core is a subset of attributes $CORE \subseteq RED$ (where $RED \subseteq P$ and $P \subseteq Q$) such that:

- It consists of attributes which *cannot be removed* from the *DS* without causing collapse of the equivalence class structure. Formally, $[t]_{IND(RED-CORE)} \neq [t]_{IND(P)}$ where the above $A \in RED$ in this case is $A \in CORE$.

A *discernibility matrix* with a symmetrical dimension $n \times n$ is constructed to compute reducts and core. n denotes the number of elementary sets and each of the matrix's elements d_{ij} is defined as the set of all attributes which discern elementary sets $[t]_{IND(Pi)}$ and $[t]_{IND(Pj)}$ [17].

3.5. Decision rules interpreted from protective relay event report

Relay *DS* analysis is considered as a supervised learning problem (classification) [13]. A *DS* determines a logical implication called *decision rule* when the conditions specified by condition attributes in each row of *DS* correlate what decisions (trip assertions) are to take effect [18]. Thus, in this study the logical implication is designated as *relay decision rule*. A complete set of relay decision rules can be derived from the relay decision table *DS*. Events in *DS*, i.e. $\{t_1, t_2, t_3, \ldots, t_m\} = U$, identify as labels of relay decision rules.

Formally, let

- $U | IND(C)$ be *condition classes* in relay *DS* (a family of all *C*-elementary sets), denoted by X_i $(i = 1, \ldots, k)$,

- $U|IND(D)$ be *decision classes* in relay *DS* (a family of all *D*-elementary sets), denoted by Y_j (j = 1, …, n).

Then, $Des_C(X_i) \Rightarrow Des_D(Y_j)$ is called relay *CD-decision rule*. For simplicity, $C \Rightarrow D$. (As aforementioned, $Des_P(X) = \{(q,v) : f(x,q) = v, \forall x \in X, \forall q \in P\}$ which denotes a description of *P*-elementary set $X \in U|IND(P)$ in terms of values of attributes from *P*).

The relay *CD*-decision rules are logical statements read as 'if *C*…then…*D*'. These rule correlate descriptions of condition attributes $C \subset Q$ (for internal multifunctional protective elements, voltages, currents and impedance measurements) to classes of decision attribute $D \subset Q$ (i.e. type of trip assertions).

The set of decision rules for *each* decision class Y_j (j = 1,…, n) is denoted by:

$$\{r_{ij}\} = \{Des_C(X_i) \Rightarrow Des_D(Y_j) : X_i \cap Y_j \neq \varnothing, i = 1, \ldots, \mathrm{k}\} \quad (10)$$

Decision algorithm in *DS* is used to mean the set of decision rules for *all* decision classes, i.e. *CD-algorithm* [10,18]. In the context of protective relay operation characteristics, a decision algorithm is a collection of relay *CD*-decision rules, thus referred to as *relay CD-decision algorithm* in this study.

Rules having the same conditions but different decisions are *inconsistent* (*nondeterministic, conflicting*); otherwise they are *consistent* (*certain, deterministic, nonconflicting*) [17]. When some conditions are satisfied, deterministic *DS uniquely* describes the decisions (actions) to be made. In a non-deterministic *DS*, decisions are not uniquely determined by the conditions [9]. Formally, it is defined that:

- Relay rule $\{r_{ij}\}$ is *deterministic* in *DS* if and only if $X_i \subseteq Y_j$, and
- Relay rule $\{r_{ij}\}$ is *nondeterministic* in *DS*, otherwise.

The *degree of consistency* (or *degree of dependency*) between the set of attributes *C* and *D* of a relay *CD*-decision algorithm is denoted as $C \Rightarrow_k D$ and can be formally defined as:

$$C \Rightarrow_{\mathrm{k}} D \mid \mathrm{k} = \gamma(C, D) = \frac{|POS_C D|}{|U|} \quad (11)$$

(i.e. conceptually similar to the quality of approximation or classification) [10]. In other words, *D depends on C* in a degree of dependency k ($0 \le k \le 1$). All the values of attributes from *D depend totally* on (i.e. uniquely determined by) the values of attributes from *C* if $k = 1$, i.e. $C \Rightarrow_1 D$ or simply $C \Rightarrow D$. *D depends partially* in a degree k on *C* if $k < 1$ [17].

It may happen that the set *D* depends on subset *C′* called *relative reduct* and not on the entire set *C*. *C′ is* a relative reduct called *D-reduct of* C if $C' \subseteq C$ is a minimal subset of *C* and $\gamma(C, D) = \gamma(C', D)$ is valid (i.e. similar in dependency). $RED_D(C)$ is used to mean the family of all *D*-reducts of *C* [18]. Putting it simply, the minimal subsets of condition attributes that discern all decision equivalence classes of the relation $U|IND(D)$ discernable by the entire set of attributes are called *D*-reducts [11]. The following notations are, thus, valid:

- If $POS_C(D) = POS_{(C-\{c_i\})}(D)$, an attribute $c_i \in C$ is *D-dispensable* in *C*. c_i is *D*-superfluous if it exerts no influence on the lower approximation of *D*. Otherwise the attribute c_i is *D-indispensable* in *C*.
- If *C* is *D-independent*, then all attributes $c_i \in C$ are *D*-indispensable in *C* and called the *D-core of* *C* which is denoted as $CORE_D(C)$.
- The following property is also true for *DS* system as previously defined,

$$CORE_D\left(C\right) \;=\; \cap RED_D\left(C\right) \tag{12}$$

  The previous definitions are valid if $D = C$ [18].
- Using a slightly modified discernibility matrix called *D-discernibility matrix of* C, relative reducts can be computed. The set of all condition attributes which discern events t_i and t_j that do not belong to the same equivalence class of the relation $U \mid IND(D)$ defines the element of *D*-discernibility matrix of *C*. The set of all single elements of the *D*-discernibility matrix of *C* is the *D*-core of C [10,11]. Rather than the ordinary reduct of C, *D*-reduct of *C* is very much the essence of this paper's study that aspires to derive the relay *CD*-decision rules (i.e. $C \Rightarrow D$).

4. Discovering decision algorithm of numerical distance protective relay

In order to fairly understand the indiscernibility relation and rules discovery from distance protective relay decision system *DS*, the following tutorial is presented.

4.1. Protective relay decision table

Table 1 illustrates an example of a decision system $DS = \langle U, Q, V, f\rangle$ excerpted from an event report of a protective distance relay. The decision table is a presentation of information function f: $U\times Q \to V$. C = {*ag*, *bg*, *cg*, *Z1pu*, *Z2pu*, *Z3pu*, *Z4pu*, *Z1trp*, *Z2trp*, *Z3trp*, *Z4trp*} is the set of condition attributes representing the internal multifunctional protective elements. D = *Trip* is the decision attribute which, essentially, denotes the tripping signal asserted by the relay in response to a particular fault in the power system. The time codes are the events that are analyzed for equivalence relation on the basis of selected subset of attributes *P*, such that $P \subseteq Q$. The finite set of the attribute *time*'s code forms the universe of interest $U = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}, t_{11}, t_{12}, t_{13}, t_{14}, t_{15}, t_{16}, t_{17}, t_{18}, t_{19}, t_{20}, t_{21}\}$.

time(U)		*ag*	*bg*	*cg*	*Z1pu*	*Z2pu*	*Z3pu*	*Z4pu*	*Z1trp*	*Z2trp*	*Z3trp*	*Z4trp*	*Trip*
sec	code	zone	zone	zone	logic	logic	logic	logic	logic	logic	logic	logic	pole
0.4982	t_1	0	0	0	0	0	0	0	0	0	0	0	0
0.4994	t_2	0	0	0	0	0	0	0	0	0	0	0	0
0.5006	t_3	0	0	0	0	0	0	0	0	0	0	0	0
0.5018	t_4	0	1	0	0	1	1	0	0	0	0	0	0
0.5030	t_5	0	2	0	0	1	1	0	0	0	0	0	0
0.5054	t_6	0	2	0	0	1	1	0	0	0	0	0	0

time(U)		*ag*	*bg*	*cg*	*Z1pu*	*Z2pu*	*Z3pu*	*Z4pu*	*Z1trp*	*Z2trp*	*Z3trp*	*Z4trp*	*Trip*
sec	code	zone	zone	zone	logic	logic	logic	logic	logic	logic	logic	logic	pole
0.5066	t_7	0	2	0	0	1	1	0	0	0	0	0	0
0.5498	t_8	0	1	0	1	1	1	0	1	0	0	0	b
0.5510	t_9	0	1	0	1	1	1	0	1	0	0	0	b
0.5522	t_{10}	0	1	0	1	1	1	0	1	0	0	0	b
0.5534	t_{11}	0	0	0	1	1	1	0	0	0	0	0	b
0.5546	t_{12}	0	0	0	1	1	1	0	0	0	0	0	b
0.5558	t_{13}	0	0	0	1	1	1	0	0	0	0	0	b
0.5966	t_{14}	0	0	0	1	1	1	0	0	0	0	0	b
0.5978	t_{15}	0	0	0	1	1	1	0	0	0	0	0	b
0.5990	t_{16}	0	0	0	0	0	1	0	0	0	0	0	b
0.6002	t_{17}	0	0	0	0	0	0	0	0	0	0	0	b
0.6014	t_{18}	0	0	0	0	0	0	0	0	0	0	0	b
0.6026	t_{19}	0	0	0	0	0	0	0	0	0	0	0	b
0.7347	t_{20}	0	0	0	0	0	0	0	0	0	0	0	0
0.7359	t_{21}	0	0	0	0	0	0	0	0	0	0	0	0

Table 1. Excerpt of an event report as a decision table *DS* of a protective distance relay (only ground distance is considered for illustration)

The attribute names are described as follows:

- *ag*, *bg,* and *cg* are A-G, B-G, and C-G fault detections.
- *Z1pu*, *Z2pu*, *Z3pu*, and *Z4pu* are zone 1, 2, 3, and 4 ground distance starts (pick-ups).
- *Z1trp*, *Z2trp*, *Z3trp*, and *Z4trp* are zone 1, 2, 3, and 4 ground distance trip signals.

The sets of values (domains) of the particular attributes are as follows:

- V_{ag}, V_{bg}, V_{cg}, = {1, 2, 3, 4}.
- V_{Z1pu}, V_{Z2pu}, V_{Z3pu}, V_{Z4pu}, V_{Z1trp}, V_{Z2trp}, V_{Z3trp}, V_{Z4trp} = {0, 1}.
- V_{Trip} = {a, b, c, 0}, corresponding to tripping signals of phase A, B, C or none.

4.2. Protective relay decision table analysis

From Table 1, the two elementary sets with respect to the decision attribute *D* = {*Trip*} can be deduced as shown in Table 2.

U/D	*Trip*
$\{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_{20}, t_{21}\} = D_1$	0
$\{t_8, t_9, t_{10}, t_{11}, t_{12}, t_{13}, t_{14}, t_{15}, t_{16}, t_{17}, t_{18}, t_{19}\} = D_2$	b

Table 2. Equivalence classes with respect to decision attribute *D* = {*Trip*}

Six equivalence classes (elementary sets) can be deduced as shown in Table 3 when the full set of attributes *C* = {*ag*, *bg*, *cg*, *Z1pu*, *Z2pu*, *Z3pu*, *Z4pu*, *Z1trp*, *Z2trp*, *Z3trp*, *Z4trp*} is considered.

U/C	*ag*	*bg*	*cg*	*Z1pu*	*Z2pu*	*Z3pu*	*Z4pu*	*Z1trp*	*Z2trp*	*Z3trp*	*Z4trp*
$\{t_1, t_2, t_3, t_{17}, t_{18}, t_{19}, t_{20}, t_{21}\}$	0	0	0	0	0	0	0	0	0	0	0
$\{t_4\}$	0	1	0	0	1	1	0	0	0	0	0
$\{t_5, t_6, t_7\}$	0	2	0	0	1	1	0	0	0	0	0
$\{t_8, t_9, t_{10}\}$	0	1	0	1	1	1	0	1	0	0	0
$\{t_{11}, t_{12}, t_{13}, t_{14}, t_{15}\}$	0	0	0	1	1	1	0	0	0	0	0
$\{t_{16}\}$	0	0	0	0	0	1	0	0	0	0	0

Table 3. Equivalence classes with respect to condition attributes $C = \{ag, bg, cg, Z1pu, Z2pu, Z3pu, Z4pu, Z1trp, Z2trp, Z3trp, Z4trp\}$

Within the first equivalence class, $\{t_1, t_2, t_3, t_{17}, t_{18}, t_{19}, t_{20}, t_{21}\}$, the eight events are indiscernible among each other based on the available attributes. In the third and the fourth equivalence classes, $\{t_5, t_6, t_7\}$ and $\{t_8, t_9, t_{10}\}$, the three events within them, based on the available attributes, cannot be distinguished from one another. Similarly, the five events within the fifth equivalence class are also indiscernible from one another. The remaining two events are each discernible (different) from all other events. $[t]_{IND(C)}$ or simply $[t]_C$ can denote these equivalence classes of the *C*-indiscernibility relation as aforementioned. Each row in Table 3 describes an individual elementary set, whereas the entire Table 3 describes the *DS* being studied. $U|C$ means that elementary sets of the universe *U* in the space *C* are being considered.

The calculations of the *C*-lower and *C*-upper approximations and accuracy of classification of *D*,

$$\underline{C}D_1 = \{t_4\} \cup \{t_5, t_6, t_7\} = \{t_4, t_5, t_6, t_7\} \tag{13}$$

$$\begin{aligned} \underline{C}D_2 &= \{t_8,\ t_9,\ t_{10}\} \cup \{t_{11},\ t_{12},\ t_{13},\ t_{14},\ t_{15}\} \cup \{t_{16}\} \\ &= \{t_8,\ t_9,\ t_{10},\ t_{11},\ t_{12},\ t_{13},\ t_{14},\ t_{15}, t_{16}\} \end{aligned} \tag{14}$$

$$\begin{aligned} \overline{C}D_1 &= \{t_1,\ t_2,\ t_3,\ t_{17},\ t_{18},\ t_{19},\ t_{20},\ t_{21}\} \cup \{t_4\} \cup \{t_5,\ t_6,\ t_7\} \\ &= \{t_1,\ t_2,\ t_3, t_4,\ t_5,\ t_6,\ t_7, t_{17},\ t_{18},\ t_{19},\ t_{20},\ t_{21}\} \end{aligned} \tag{15}$$

$$\begin{aligned} \overline{C}D_2 &= \{t_1,\ t_2,\ t_3,\ t_{17},\ t_{18},\ t_{19},\ t_{20},\ t_{21}\} \cup \{t_8,\ t_9,\ t_{10}\} \cup \{t_{11},\ t_{12},\ t_{13},\ t_{14},\ t_{15}\} \cup \{t_{16}\} \\ &= \{t_1,\ t_2,\ t_3, t_8,\ t_9,\ t_{10},\ t_{11},\ t_{12},\ t_{13},\ t_{14},\ t_{15}, t_{16},\ t_{17},\ t_{18},\ t_{19},\ t_{20},\ t_{21}\} \end{aligned} \tag{16}$$

$$\alpha_C(D_1) = \left|\frac{\underline{C}D_1}{\overline{C}D_1}\right| = \frac{card(\underline{C}D_1)}{card(\overline{C}D_1)} = \frac{4}{12} = 0.33 \tag{17}$$

$$\alpha_C(D_2) = \left|\frac{\underline{C}D_2}{\overline{C}D_2}\right| = \frac{card(\underline{C}D_2)}{card(\overline{C}D_2)} = \frac{9}{17} = 0.53 \tag{18}$$

With classification accuracies of 0.33 and 0.53, the respective elementary sets D_1 and D_2 are roughly definable (vaguely classified) in the *DS*. This is rather expected. The decision attribute

D = {*Trip*} may remain in a certain domain value for a certain time-sequence of relay events after a particular relay trip trigger according to the protection engineer's preset time duration of signal assertion [7]. This may prevail even though the condition attributes have changed during this duration. This explains the inconsistency found in the *CD*-algorithm.

The accuracy and quality of overall classification D are:

$$\alpha_C(D) = \frac{\sum_{i=1}^{2} card(\underline{C}D_i)}{\sum_{i=1}^{2} card(\overline{C}D_i)} = \frac{4+9}{12+17} = 0.45 \tag{19}$$

$$\gamma_C(D) = \frac{\sum_{i=1}^{2} card(\underline{C}D_i)}{card(U)} = \frac{4+9}{9+12} = 0.62 \tag{20}$$

i.e. the overall classification with respect to C is rough.

D-reducts and D-core of C can be discovered from the D-discernibility matrix of C by discerning relay events from different equivalence classes in the relation $U|IND(D)$ with respect to the condition attributes C. The D-discernibility matrix that is formed is illustrated in Table 4. It would suffice to consider only the lower diagonal part because of the matrix's symmericalness [11]. Note that even though relay events appearing in the same class in the D-space (for example t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_{20}, t_{21}) are discernible in C-space, they are not discerned between each other with respect to the attributes C. Empty set (Ø) indicates indiscernibility between relay events.

U	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}	t_{13}	t_{14}	t_{15}	t_{16}	t_{17}	t_{18}	t_{19}	t_{20}	t_{21}
t_1	Ø																				
t_2	Ø	Ø																			
t_3	Ø	Ø	Ø																		
t_4	Ø	Ø	Ø	Ø																	
t_5	Ø	Ø	Ø	Ø	Ø																
t_6	Ø	Ø	Ø	Ø	Ø	Ø															
t_7	Ø	Ø	Ø	Ø	Ø	Ø	Ø														
t_8	{*bg, Z1pu, Z2pu, Z3pu, Z1trp*}	{*bg, Z1pu, Z2pu, Z3pu, Z1trp*}	{*bg, Z1pu, Z2pu, Z3pu, Z1trp*}	{*Z1pu, Z1trp*}	{*bg, Z1pu, Z1trp*}	{*bg, Z1pu, Z1trp*}	{*bg, Z1pu, Z1trp*}	Ø													
t_9	{*bg, Z1pu, Z2pu, Z3pu, Z1trp*}	{*bg, Z1pu, Z2pu, Z3pu, Z1trp*}	{*bg, Z1pu, Z2pu, Z3pu, Z1trp*}	{*Z1pu, Z1trp*}	{*bg, Z1pu, Z1trp*}	{*bg, Z1pu, Z1trp*}	{*bg, Z1pu, Z1trp*}	Ø	Ø												
t_{10}	{*bg, Z1pu, Z2pu, Z3pu, Z1trp*}	{*bg, Z1pu, Z2pu, Z3pu, Z1trp*}	{*bg, Z1pu, Z2pu, Z3pu, Z1trp*}	{*Z1pu, Z1trp*}	{*bg, Z1pu, Z1trp*}	{*bg, Z1pu, Z1trp*}	{*bg, Z1pu, Z1trp*}	Ø	Ø	Ø											
t_{11}	{*Z1pu, Z2pu, Z3pu*}	{*Z1pu, Z2pu, Z3pu*}	{*Z1pu, Z2pu, Z3pu*}	{*bg, Z1pu*}	{*bg, Z1pu*}	{*bg, Z1pu*}	{*bg, Z1pu*}	Ø	Ø	Ø	Ø										

U	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}	t_{13}	t_{14}	t_{15}	t_{16}	t_{17}	t_{18}	t_{19}	t_{20}	t_{21}
t_{12}	{Z1pu, Z2pu, Z3pu}	{Z1pu, Z2pu, Z3pu}	{Z1pu, Z2pu, Z3pu}	{bg, Z1pu}	{bg, Z1pu}	{bg, Z1pu}	{bg, Z1pu}	∅	∅	∅	∅	∅									
t_{13}	{Z1pu, Z2pu, Z3pu}	{Z1pu, Z2pu, Z3pu}	{Z1pu, Z2pu, Z3pu}	{bg, Z1pu}	{bg, Z1pu}	{bg, Z1pu}	{bg, Z1pu}	∅	∅	∅	∅	∅	∅								
t_{14}	{Z1pu, Z2pu, Z3pu}	{Z1pu, Z2pu, Z3pu}	{Z1pu, Z2pu, Z3pu}	{bg, Z1pu}	{bg, Z1pu}	{bg, Z1pu}	{bg, Z1pu}	∅	∅	∅	∅	∅	∅	∅							
t_{15}	{Z1pu, Z2pu, Z3pu}	{Z1pu, Z2pu, Z3pu}	{Z1pu, Z2pu, Z3pu}	{bg, Z1pu}	{bg, Z1pu}	{bg, Z1pu}	{bg, Z1pu}	∅	∅	∅	∅	∅	∅	∅	∅						
t_{16}	{Z3pu}	{Z3pu}	{Z3pu}	{bg, Z2pu}	{bg, Z2pu}	{bg, Z2pu}	{bg, Z2pu}	∅	∅	∅	∅	∅	∅	∅	∅	∅					
t_{17}	∅	∅	∅	{bg, Z2pu, Z3pu}	{bg, Z2pu, Z3pu}	{bg, Z2pu, Z3pu}	{bg, Z2pu, Z3pu}	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅				
t_{18}	∅	∅	∅	{bg, Z2pu, Z3pu}	{bg, Z2pu, Z3pu}	{bg, Z2pu, Z3pu}	{bg, Z2pu, Z3pu}	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅			
t_{19}	∅	∅	∅	{bg, Z2pu, Z3pu}	{bg, Z2pu, Z3pu}	{bg, Z2pu, Z3pu}	{bg, Z2pu, Z3pu}	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅		
t_{20}	∅	∅	∅	∅	∅	∅	∅	{bg, Z1pu, Z2pu, Z3pu, Z1trp}	{bg, Z1pu, Z2pu, Z3pu, Z1trp}	{bg, Z1pu, Z2pu, Z3pu, Z1trp}	{Z1pu, Z2pu, Z3pu}	{Z1pu, Z2pu, Z3pu}	{Z1pu, Z2pu, Z3pu}	{Z1pu, Z2pu, Z3pu}	{Z1pu, Z2pu, Z3pu}	{Z3pu}	∅	∅	∅	∅	
t_{21}	∅	∅	∅	∅	∅	∅	∅	{bg, Z1pu, Z2pu, Z3pu, Z1trp}	{bg, Z1pu, Z2pu, Z3pu, Z1trp}	{bg, Z1pu, Z2pu, Z3pu, Z1trp}	{Z1pu, Z2pu, Z3pu}	{Z1pu, Z2pu, Z3pu}	{Z1pu, Z2pu, Z3pu}	{Z1pu, Z2pu, Z3pu}	{Z1pu, Z2pu, Z3pu}	{Z3pu}	∅	∅	∅	∅	∅

Table 4. *D*-discernibility matrix of *C*

The discovery of the desired reduct(s) is possible via the formulation of the so-called *discernibility function f(P)* that calculates according to Boolean function operation in which each attribute acts as a Boolean variable [11]. Using the technique introduced by Pawlak [17], a Boolean discernibility function is deduced right off the discernibility matrix in Table 4, i.e.:

$f_C(D)$

$= (bg+Z1pu+Z2pu+Z3pu+Z1trp)\ (Z1pu+Z2pu+Z3pu)\ (Z3pu) \times (Z1pu+Z1trp)\ (bg+Z1trp)\ (bg+Z2pu)\ (bg+Z2pu+Z3pu) \times (bg+Z1pu+Z1trp)\ (bg+Z1pu)\ (bg+Z2pu)\ (bg+Z2pu+Z3pu) \times (bg+Z1pu+Z2pu+Z3pu+Z1trp) \times (Z1pu+Z2pu+Z3pu) \times Z3pu$

$= Z3pu \times (Z1pu+Z1trp)\ (bg+Z1pu)\ (bg+Z2pu) \times (bg+Z1pu)\ (bg+Z2pu) \times (bg+Z1pu+Z2pu+Z3pu+Z1trp) \times (Z1pu+Z2pu+Z3pu) \times Z3pu \rightarrow$ The final Conjunctive Normal Form (CNF)

$= Z3pu\ (Z1pu+Z1trp)\ (bg+Z1pu)\ (bg+Z2pu)$

$= (Z1pu{\cdot}Z3pu + Z3pu{\cdot}Z1trp)\ (bg+Z1pu)\ (bg+Z2pu)$

$= (bg{\cdot}(Z1pu{\cdot}Z3pu + Z3pu{\cdot}Z1trp) + Z1pu(Z1pu{\cdot}Z3pu + Z3pu{\cdot}Z1trp))\ (bg+Z2pu)$

$= ((bg{\cdot}Z1pu{\cdot}Z3pu + bg{\cdot}Z3pu{\cdot}Z1trp) + (Z1pu{\cdot}Z3pu) + (Z1pu{\cdot}Z3pu{\cdot}Z1trp))\ (bg+Z2pu)$

= *bg* ((*bg·Z1pu·Z3pu* + *bg·Z3pu·Z1trp*) + (*Z1pu·Z3pu*) + (*Z1pu·Z3pu·Z1trp*)) + *Z2pu* ((*bg·Z1pu·Z3pu* + *bg·Z3pu·Z1trp*) + (*Z1pu·Z3pu*) + (*Z1pu·Z3pu·Z1trp*))

= (*bg·Z1pu·Z3pu*) + (*bg·Z3pu·Z1trp*) + (*bg·Z1pu·Z3pu*) + (*bg·Z1pu·Z3pu·Z1trp*) + (*bg·Z1pu·Z2pu· Z3pu*) + (*bg·Z2pu·Z3pu·Z1trp*) + (*Z1pu·Z2pu·Z3pu*) + (*Z1pu·Z2pu·Z3pu·Z1trp*)

→ The final Disjunctive Normal Form (DNF) of $f_C(D)$

Absorption law and eventual expression multiplication are implemented to solve the Boolean expression of $f_C(D)$ [19].

Normalization in its final normal form, the last Boolean expression $f_C(D)$ is recognized as Disjunctive Normal Form (DNF). DNF is analogous to Sum Of Product (SOP) boolean algebra in digital electronics logic. $f_C(D)$ in DNF form is an alternative representation of the *DS* in which all its constituents are the *D*-reducts of *C* (i.e. $RED_D(C)$) [11,17]. Either one of the set of reducts can be used to represent exactly the same data classification as that depicted by the entire set of attributes *C*. The following $RED_D(C)$ of the above final $f_C(D)$ reveals that either one of the *D*-reducts of *C* can be used alternatively to represent exactly the same equivalence relation $U|IND(D)$ of the *DS* as that represented by the whole set of attributes *C*, i.e.,

$$\begin{gathered} RED_D(C) = \{bg, Z1pu, Z3pu\}, \{bg, Z3pu, Z1trp\}, \{bg, Z1pu, Z3pu\}, \\ \{bg, Z1pu, Z3pu, Z1trp\}, \{bg, Z1pu, Z2pu, Z3pu\}, \{bg, Z2pu, Z3pu, Z1trp\}, \\ \{Z1pu, Z2pu, Z3pu\}, \{Z1pu, Z2pu, Z3pu, Z1trp\} \end{gathered} \tag{21}$$

The *D*-core of *C* can be figured out by either:

- Identifying all the single attribute entries in the *D*-discernibility matrix of C [11], which from Table 4, attribute *Z3pu* is the only single attribute entry and thus $CORE_D(C) = \cap RED_D(C) = Z3p$, or
- Taking intersection of all *D*-reducts of *C*, i.e. $CORE_D(C) = \cap RED_D(C) = Z3pu$

Hence, *Z3pu* is the most characteristic attribute that is indispensible in *DS* without reducing the approximation quality of equivalence relation $U|C$ with respect to *D*.

$CORE_D(C) = Z3pu$ does not seem to signify any significance in the behavior of the relay under analysis. Had the reduct analysis been worked out based only on the whole condition attributes *C* (as per the equivalence relation in Table 3, where decision attribute *D* is excluded such as in the case of *IS* instead of *DS*), the core of *C* (i.e. the core of the equivalence relation $U|C$ with respect to C) would have been,

$$CORE_{U|C}(C) = \{bg\} \tag{22}$$

This implies the protective relay has been subjected to B-G fault. In reality this fault occurred in distance zone 1 operation characteristic and was picked up by the zone 1 distance element. However, the *D*-core of *C* discovers the indispensability of the condition attribute *Z3pu* as being the core when the decision attribute *D* is considered for the *DS* analysis. Actually, this attribute is entirely insignificant based on the understanding of the manner the distance relay functions. This is simply because of the concurrent nature of the distance relay quadrilateral operation characteristic whereby zone 1 element is encapsulated in zone 2 element and subsequently zone 2 element is encapsulated in zone 3 element. Zone 4 element is on its own separate entity not encapsulated in any zone elements [7]. Thus, by merely considering the exertion of the zone 1 element in case of fault and correspondingly disregarding zones 2, 3 and 4 operation is principally correct. Figure 2 illustrates that a fault occurring in zone 1 is also concurrently shown as present in zones 2 and 3 as well.

To simplify and make the analysis process more sense, an attribute priority of the distance relay operation has to be formulated so that the relay *DS* can be modified as shown Table 5.

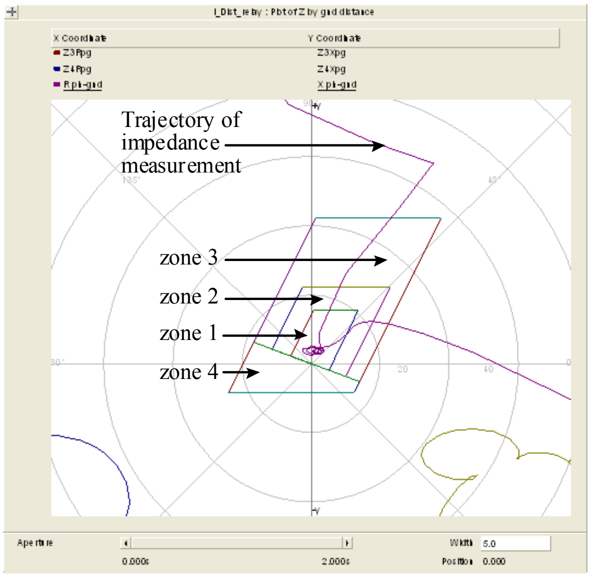

Figure 2. Distance protective relay operation characteristic with impedance measurement

Cases of concurrence	Condition Attributes, $c_i \in C$								
	Z1pu	*Z2pu*	*Z3pu*	*Z4pu*	*Z1trp*	*Z2trp*	*Z3trp*	*Z4trp*	Most significant attribute
Case 1	+	+	+		+				*Z1pu*
Case 1′	+	+	+			*			*Z2pu*
Case 2		+	+			*			*Z2pu*
Case 2′		+	+				*		*Z3pu*
Case 3			+				*		*Z3pu*
Case 4				+				*	*Z4pu*

+ denotes value of attribute equal to "1", i.e. $V_{ci} = 1$ where attribute $c_i \in C$.
* denotes the attribute's value of "1" occurring at possibly different events (rows).

Table 5. Condition attribute priority of the distance relay operation

The absence of relay trip assertion signal in attributes *Z2trp*, *Z3trp*, and *Z4trp* which is represented by the attribute value "0" further justifies the necessity of disregarding attributes *Z2pu*, *Z3pu*, and *Z4pu* for fault in zone 1. This is because, for example, the assertion of attribute *Z1pu* (value of "1") must always be accompanied by the assertion (after and for a preset time duration, i.e. sequence of consecutive events) of the corresponding attribute *Z1trp* in order to be taken into consideration in the analysis. However, in the above example, it is highly likely that attribute *Z2trp* will assert (after and for a preset number of events) in lieu of the attribute *Z1trp* as shown in Table 5 if the relay failed to operate in asserting the attribute *Z1trp* when the attribute *Z1pu* is asserted.

Taking into account the proposition, the *DS* system in Table 1 is then modified prior to reanalysis using rough set as shown in Table 6.

Time (*U*)	*ag*	*bg*	*cg*	*Z1pu*	*Z4pu*	*Z1trp*	*Z2trp*	*Z3trp*	*Z4trp*	*Trip*
t_1	0	0	0	0	0	0	0	0	0	0
t_2	0	0	0	0	0	0	0	0	0	0
t_3	0	0	0	0	0	0	0	0	0	0
t_4	0	1	0	0	0	0	0	0	0	0
t_5	0	2	0	0	0	0	0	0	0	0
t_6	0	2	0	0	0	0	0	0	0	0
t_7	0	2	0	0	0	0	0	0	0	0
t_8	0	1	0	1	0	1	0	0	0	b
t_9	0	1	0	1	0	1	0	0	0	b
t_{10}	0	1	0	1	0	1	0	0	0	b
t_{11}	0	0	0	1	0	0	0	0	0	b
t_{12}	0	0	0	1	0	0	0	0	0	b
t_{13}	0	0	0	1	0	0	0	0	0	b
t_{14}	0	0	0	1	0	0	0	0	0	b
t_{15}	0	0	0	1	0	0	0	0	0	b
t_{16}	0	0	0	0	0	0	0	0	0	b
t_{17}	0	0	0	0	0	0	0	0	0	b
t_{18}	0	0	0	0	0	0	0	0	0	b
t_{19}	0	0	0	0	0	0	0	0	0	b
t_{20}	0	0	0	0	0	0	0	0	0	0
t_{21}	0	0	0	0	0	0	0	0	0	0

Table 6. Modified decision table *DS* to reflect protective relay operation behavior

From Table 6, the elementary sets with respect to the decision attribute *D* = {*Trip*} are still the same as shown in Table 2.

However, the elementary sets with respect to the shrunk condition = {*ag*, *bg*, *cg*, *Z1pu*, *Z4pu*, *Z1trp*, *Z2trp*, *Z3trp*, *Z4trp*} as shown in Table 7 are slightly different from those found with the whole attributes *C* considered (Table 3).

U/C	*ag*	*bg*	*cg*	*Z1pu*	*Z4pu*	*Z1trp*	*Z2trp*	*Z3trp*	*Z4trp*
{t_1, t_2, t_3, t_{16}, t_{17}, t_{18}, t_{19}, t_{20}, t_{21}}	0	0	0	0	0	0	0	0	0
{t_4}	0	1	0	0	0	0	0	0	0
{t_5, t_6, t_7}	0	2	0	0	0	0	0	0	0
{t_8, t_9, t_{10}}	0	1	0	1	0	1	0	0	0
{t_{11}, t_{12}, t_{13}, t_{14}, t_{15}}	0	0	0	1	0	0	0	0	0

Table 7. Equivalence classes with respect to modified condition attributes C = {*ag*, *bg*, *cg*, *Z1pu*, *Z4pu*, *Z1trp*, *Z2trp*, *Z3trp*, *Z4trp*}

The new D-discernibility matrix of C as in Table 8 will result in new D-reducts and D-core of C when events are discerned with respect to the modified condition attributes C between different equivalence classes in the relation $U|IND(D)$. As before, similar consideration is taken in discerning events appearing only in different classes in D-space.

U	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}	t_{13}	t_{14}	t_{15}	t_{16}	t_{17}	t_{18}	t_{19}	t_{20}	t_{21}
t_1	Ø																				
t_2	Ø	Ø																			
t_3	Ø	Ø	Ø																		
t_4	Ø	Ø	Ø	Ø																	
t_5	Ø	Ø	Ø	Ø	Ø																
t_6	Ø	Ø	Ø	Ø	Ø	Ø															
t_7	Ø	Ø	Ø	Ø	Ø	Ø	Ø														
t_8	{*bg*, *Z1pu*, *Z1trp*}	{*bg*, *Z1pu*, *Z1trp*}	{*bg*, *Z1pu*, *Z1trp*}	{*Z1pu*, *Z1trp*}	{*bg*, *Z1pu*, *Z1trp*}	{*bg*, *Z1pu*, *Z1trp*}	{*bg*, *Z1pu*, *Z1trp*}	Ø													
t_9	{*bg*, *Z1pu*, *Z1trp*}	{*bg*, *Z1pu*, *Z1trp*}	{*bg*, *Z1pu*, *Z1trp*}	{*Z1pu*, *Z1trp*}	{*bg*, *Z1pu*, *Z1trp*}	{*bg*, *Z1pu*, *Z1trp*}	{*bg*, *Z1pu*, *Z1trp*}	Ø	Ø												
t_{10}	{*bg*, *Z1pu*, *Z1trp*}	{*bg*, *Z1pu*, *Z1trp*}	{*bg*, *Z1pu*, *Z1trp*}	{*Z1pu*, *Z1trp*}	{*bg*, *Z1pu*, *Z1trp*}	{*bg*, *Z1pu*, *Z1trp*}	{*bg*, *Z1pu*, *Z1trp*}	Ø	Ø	Ø											
t_{11}	{*Z1pu*}	{*Z1pu*}	{*Z1pu*}	{*bg*, *Z1pu*}	{*bg*, *Z1pu*}	{*bg*, *Z1pu*}	{*bg*, *Z1pu*}	Ø	Ø	Ø	Ø										
t_{12}	{*Z1pu*}	{*Z1pu*}	{*Z1pu*}	{*bg*, *Z1pu*}	{*bg*, *Z1pu*}	{*bg*, *Z1pu*}	{*bg*, *Z1pu*}	Ø	Ø	Ø	Ø	Ø									
t_{13}	{*Z1pu*}	{*Z1pu*}	{*Z1pu*}	{*bg*, *Z1pu*}	{*bg*, *Z1pu*}	{*bg*, *Z1pu*}	{*bg*, *Z1pu*}	Ø	Ø	Ø	Ø	Ø	Ø								
t_{14}	{*Z1pu*}	{*Z1pu*}	{*Z1pu*}	{*bg*, *Z1pu*}	{*bg*, *Z1pu*}	{*bg*, *Z1pu*}	{*bg*, *Z1pu*}	Ø	Ø	Ø	Ø	Ø	Ø	Ø							
t_{15}	*Z1pu*}	*Z1pu*}	*Z1pu*}	{*bg*, *Z1pu*}	{*bg*, *Z1pu*}	{*bg*, *Z1pu*}	{*bg*, *Z1pu*}	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø						
t_{16}	Ø	Ø	Ø	{*bg*}	{*bg*}	{*bg*}	{*bg*}	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø					
t_{17}	Ø	Ø	Ø	{*bg*}	{*bg*}	{*bg*}	{*bg*}	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø				
t_{18}	Ø	Ø	Ø	{*bg*}	{*bg*}	{*bg*}	{*bg*}	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø			
t_{19}	Ø	Ø	Ø	{*bg*}	{*bg*}	{*bg*}	{*bg*}	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø		
t_{20}	Ø	Ø	Ø	Ø	Ø	Ø	Ø	{*bg*, *Z1pu*, *Z1trp*}	{*bg*, *Z1pu*, *Z1trp*}	{*bg*, *Z1pu*, *Z1trp*}	{*Z1pu*}	{*Z1pu*}	{*Z1pu*}	{*Z1pu*}	{*Z1pu*}	Ø	Ø	Ø	Ø	Ø	
t_{21}	Ø	Ø	Ø	Ø	Ø	Ø	Ø	{*bg*, *Z1pu*, *Z1trp*}	{*bg*, *Z1pu*, *Z1trp*}	{*bg*, *Z1pu*, *Z1trp*}	{*Z1pu*}	{*Z1pu*}	{*Z1pu*}	{*Z1pu*}	{*Z1pu*}	Ø	Ø	Ø	Ø	Ø	Ø

Table 8. D-discernibility matrix of modified C

The Boolean discernibility function is formulated from the discernibility matrix as follows:

$f_C(D)$

= $(bg+Z1pu+Z1trp)\ (Z1pu) \times (Z1pu+Z1trp)\ (bg+Z1pu)\ (bg) \times (bg+Z1pu+Z1trp)\ (\mathrm{bg}+Z1pu)\ (bg) \times (bg+Z1pu+Z1trp) \times Z1pu$

= $(Z1pu) \times (Z1pu+Z1trp)\ (bg) \times (bg) \times (bg+Z1pu+Z1trp) \times Z1pu$

= $(Z1pu)\ (bg) \rightarrow$ The final Disjunctive Normal Form (DNF) of $f_C(D)$

There is only one D-reduct of C, $RED_D(C) = \{bg, Z1pu\}$. As shown in Table 9, it can alternatively be used to represent exactly similar equivalence relation $U|IND(D)$ of the down scaled DS as that represented by the whole set of attributes C. The D-core of C is the set of all single entries of the D-discernibility matrix, (or $CORE_D(C) = \cap RED_D(C)$), i.e. $\{bg, Z1pu\}$. In this case, the D-core of C is similar to D-reduct of C.

As previously discussed, the possibility of the core inferring the power system state the relay has been subjected to is really prominently singled out now by the new $CORE_D(C) = \{bg, Z1pu\}$. Due the very characteristic of indispensability of core, it is undoubtedly identified that a A-G fault has occurred and consequently the relay's Z1 ground distance element has picked up to get rid of it. This eventually translates into the trip decision having patterns such as that presented by the attribute *Trip* shown all along.

Time (U)	bg	$Z1pu$	$Trip$
t_1	0	0	0
t_2	0	0	0
t_3	0	0	0
t_4	1	0	0
t_5	2	0	0
t_6	2	0	0
t_7	2	0	0
t_8	1	1	b
t_9	1	1	b
t_{10}	1	1	b
t_{11}	0	1	b
t_{12}	0	1	b
t_{13}	0	1	b
t_{14}	0	1	b
t_{15}	0	1	b
t_{16}	0	0	b
t_{17}	0	0	b
t_{18}	0	0	b
t_{19}	0	0	b
t_{20}	0	0	0
t_{21}	0	0	0

Table 9. Equivalent decision table with respect to $RED_D(C) = \{bg, Z1pu\}$

4.3. Protective relay decision algorithm discovery

As aforementioned, a relay decision algorithm in *DS* called *CD*-decision algorithm manifests as a *CD*-decision table. It comprises a finite set of relay *CD*-decision rules or instructions. The event report of a protective distance relay in the form of a *DS* is a manifestation of relay decision algorithm. In protection system, protection engineers relate relay decision algorithm as relay operation logic. It is envisaged that with rough set theory, the relay operation logic knowledge can be discovered. Later it can be transformed into a knowledge base of a decision support system for determining anticipated relay behavior out of a new test *DS* [7].

Checking whether or not all the relay operation logics (decision rules) are true would enable us to check whether or not a relay decision algorithm is consistent. As aforementioned, consistency is measured by the degree of dependency *k* (or alternatively, dependency is measured by the degree of consistency) [10]. Thus, it is well understood that with the degree of consistency given in Equation (10),

$$k = \frac{card\,POS(C,D)}{card\ (CD - decision\,algorithm)} \tag{23}$$

a relay *CD*-decision algorithm has a degree *k*, i.e. the degree of dependency between condition attributes $C = \{ag, bg, cg, Z1pu, Z4pu, Z1trp, Z2trp, Z3trp, Z4trp\}$ and decision attributes $D = \{Trip\}$.

The relay *CD*-decision rules ($C \rightarrow D$) are:

rule 1: $ag_0\ bg_0\ cg_0\ Z1pu_0\ Z4pu_0\ Z1trp_0\ Z2trp_0\ Z3trp_0\ Z4trp_0 \rightarrow Trip_0$

rule 2: $ag_0\ bg_0\ cg_0\ Z1pu_0\ Z4pu_0\ Z1trp_0\ Z2trp_0\ Z3trp_0\ Z4trp_0 \rightarrow Trip_0$

rule 3: $ag_0\ bg_0\ cg_0\ Z1pu_0\ Z4pu_0\ Z1trp_0\ Z2trp_0\ Z3trp_0\ Z4trp_0 \rightarrow Trip_0$

rule 4: $ag_0\ bg_1\ cg_0\ Z1pu_0\ Z4pu_0\ Z1trp_0\ Z2trp_0\ Z3trp_0\ Z4trp_0 \rightarrow Trip_0$

rule 5: $ag_0\ bg_2\ cg_0\ Z1pu_0\ Z4pu_0\ Z1trp_0\ Z2trp_0\ Z3trp_0\ Z4trp_0 \rightarrow Trip_0$

rule 6: $ag_0\ bg_2\ cg_0\ Z1pu_0\ Z4pu_0\ Z1trp_0\ Z2trp_0\ Z3trp_0\ Z4trp_0 \rightarrow Trip_0$

rule 7: $ag_0\ bg_2\ cg_0\ Z1pu_0\ Z4pu_0\ Z1trp_0\ Z2trp_0\ Z3trp_0\ Z4trp_0 \rightarrow Trip_0$

rule 8: $ag_0\ bg_1\ cg_0\ Z1pu_1\ Z4pu_0\ Z1trp_1\ Z2trp_0\ Z3trp_0\ Z4trp_0 \rightarrow Trip_b$

rule 9: $ag_0\ bg_1\ cg_0\ Z1pu_1\ Z4pu_0\ Z1trp_1\ Z2trp_0\ Z3trp_0\ Z4trp_0 \rightarrow Trip_b$

rule 10: $ag_0\ bg_1\ cg_0\ Z1pu_1\ Z4pu_0\ Z1trp_1\ Z2trp_0\ Z3trp_0\ Z4trp_0 \rightarrow Trip_b$

rule 11: $ag_0\ bg_0\ cg_0\ Z1pu_1\ Z4pu_0\ Z1trp_0\ Z2trp_0\ Z3trp_0\ Z4trp_0 \rightarrow Trip_b$

rule 12: $ag_0\ bg_0\ cg_0\ Z1pu_1\ Z4pu_0\ Z1trp_0\ Z2trp_0\ Z3trp_0\ Z4trp_0 \rightarrow Trip_b$

rule 13: $ag_0\ bg_0\ cg_0\ Z1pu_1\ Z4pu_0\ Z1trp_0\ Z2trp_0\ Z3trp_0\ Z4trp_0 \rightarrow Trip_b$

rule 14: $ag_0\ bg_0\ cg_0\ Z1pu_1\ Z4pu_0\ Z1trp_0\ Z2trp_0\ Z3trp_0\ Z4trp_0 \rightarrow Trip_b$

rule 15: $ag_0\ bg_0\ cg_0\ Z1pu_1\ Z4pu_0\ Z1trp_0\ Z2trp_0\ Z3trp_0\ Z4trp_0 \rightarrow Trip_b$

rule 16: $ag_0\ bg_0\ cg_0\ Z1pu_0\ Z4pu_0\ Z1trp_0\ Z2trp_0\ Z3trp_0\ Z4trp_0 \rightarrow Trip_b$

rule 17: $ag_0\ bg_0\ cg_0\ Z1pu_0\ Z4pu_0\ Z1trp_0\ Z2trp_0\ Z3trp_0\ Z4trp_0 \rightarrow Trip_b$

rule 18: $ag_0\ bg_0\ cg_0\ Z1pu_0\ Z4pu_0\ Z1trp_0\ Z2trp_0\ Z3trp_0\ Z4trp_0 \rightarrow Trip_b$

rule 19: $ag_0\ bg_0\ cg_0\ Z1pu_0\ Z4pu_0\ Z1trp_0\ Z2trp_0\ Z3trp_0\ Z4trp_0 \rightarrow Trip_b$

rule 20: $ag_0\ bg_0\ cg_0\ Z1pu_0\ Z4pu_0\ Z1trp_0\ Z2trp_0\ Z3trp_0\ Z4trp_0 \rightarrow Trip_0$

rule 21: $ag_0\ bg_0\ cg_0\ Z1pu_0\ Z4pu_0\ Z1trp_0\ Z2trp_0\ Z3trp_0\ Z4trp_0 \rightarrow Trip_0$

The two sets of relay decision rules, i.e. rules 1, 2, 3 and rules 16, 17, 18, 19, altogether totaling 7 rules, are inconsistent (false). The positive region of the *CD*-decision algorithm consists of only consistent decision rules 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 20 and 21 (i.e. *card* $POS(C,D) = 14$) and, hence, the degree of dependency is $k = 14/21 = 0.67$. Since there are decision rules in the algorithm that are consistent only by the degree of 0.67 (i.e. false), the relay *CD*-decision algorithm is said to be inconsistent. The decision classes are not all uniquely discernible by conditions of all decision rules in the *CD*-decision algorithm. In other words, there are at least two decision rules having the same conditions but different implications in the decision. This phenomenon is certainly anticipated especially as shown by rules 16, 17, 18, and 19 whereby the decision attribute *Trip* remains in the value "*b*" reflecting the actual distance relay operation behavior. Technically speaking, irrespective of the presence or otherwise of the fault (assertion via attribute "*bg*") and zone 1 element pick-up (assertion via attribute "*Z1pu*"), the relay trip signal remains asserted for a certain preset duration of time [7].

4.4. Protective relay decision algorithm simplification

Algorithm reduction results in simplification of the *CD*-decision algorithm. This is done by investigating whether all condition attributes are necessary to make decisions. Therefore, reducing *CD*-decision algorithm is essentially closely related to the previous discussion on reducing *DS*.

The subset of condition attributes $C' \subseteq C$ is called a reduct of C in the *CD*-decision algorithm if the *C′D*-decision algorithm is independent and consistent, i.e. $POS(C',D) = POS(C,D)$. Therefore, the following terms are valid:

- *C′D*-decision algorithm is reduct of *CD*-decision algorithm.
- The set of all reducts of *CD*-decision algorithm is called $RED(C,D)$
- The set of all indispensible condition attributes in the *CD*-decision algorithm is called the core of the of the *CD*-decision algorithm and, similarly like before, takes on the expression, $CORE(C,D) = \cap RED(C,D)$. (In principle it is similar to the expression $CORE_D(C) = \cap RED_D(C)$).

The modified *DS* in Table 6 had found its only reduct of condition attributes, $RED(C,D) = RED_D(C) = \{bg, Z1pu\}$, in the relay *CD*-decision algorithm. The core had a similar set as that of the reduct, i.e. $CORE(C,D) = \cap RED(C,D) = \{bg, Z1pu\}$. The resulting equivalent *DS* with respect to $RED(C,D) = \{bg, Z1pu\}$ in Table 9 produces a rather simplified version of relay *CD*-decision algorithm, i.e.,

rule 1: $bg_0\ Z1pu_0 \rightarrow Trip_0$

rule 2: $bg_0\ Z1pu_0 \rightarrow Trip_0$

rule 3: $bg_0\ Z1pu_0 \rightarrow Trip_0$

rule 4: $bg_1\ Z1pu_0 \rightarrow Trip_0$

rule 5: $bg_2\ Z1pu_0 \rightarrow Trip_0$

rule 6: $bg_2\ Z1pu_0 \rightarrow Trip_0$

rule 7: $bg_2\ Z1pu_0 \rightarrow Trip_0$

rule 8: $bg_1\ Z1pu_1 \rightarrow Trip_b$

rule 9: $bg_1\ Z1pu_1 \rightarrow Trip_b$

rule 10: $bg_1\ Z1pu_1 \rightarrow Trip_b$

rule 11: $bg_0\ Z1pu_1 \rightarrow Trip_b$

rule 12: $bg_0\ Z1pu_1 \rightarrow Trip_b$

rule 13: $bg_0\ Z1pu_1 \rightarrow Trip_b$

rule 14: $bg_0\ Z1pu_1 \rightarrow Trip_b$

rule 15: $bg_0\ Z1pu_1 \rightarrow Trip_b$

rule 16: $bg_0\ Z1pu_0 \rightarrow Trip_b$

rule 17: $bg_0\ Z1pu_0 \rightarrow Trip_b$

rule 18: $bg_0\ Z1pu_0 \rightarrow Trip_b$

rule 19: $bg_0\ Z1pu_0 \rightarrow Trip_b$

rule 20: $bg_0\ Z1pu_0 \rightarrow Trip_0$

rule 21: $bg_0\ Z1pu_0 \rightarrow Trip_0$

Each relay *CD*-decision rule designation corresponds to the row label in the *DS;* for example, rule 9 corresponding to row label t_9.

The relay *CD*-decision algorithm can be cut down by removing duplicate relay *CD*-decision rules,

rule 1′ (1, 2, 3, 20, 21): $bg_0\ Z1pu_0 \rightarrow Trip_0$

rule 2′ (4): $bg_1\ Z1pu_0 \rightarrow Trip_0$

rule 3′ (5, 6, 7): $bg_2\ Z1pu_0 \rightarrow Trip_0$

rule 4′ (8, 9, 10): $bg_1\ Z1pu_1 \rightarrow Trip_b$

rule 5′ (11, 12, 13, 14, 15): $bg_0\ Z1pu_1 \rightarrow Trip_b$

rule 6′ (16, 17, 18, 19): $bg_0\ Z1pu_0 \rightarrow Trip_b$

From the apparently inconsistent rules 1′ and 6′, i.e. similar conditions but dissimilar decisions, the simplified relay *CD*-decision algorithm reveals pronouncedly its inconsistent nature. This inconsistency may not be desirable in some information system analysis. However, in as far as protective relay operation is concerned, it is interesting to know, among others:

- the time delay between relay pick-up and relay trip signal assertion – traced by identifying the translated time sequence (*DS* row label) from rule 4 (t_4) to rule 8 (t_8),
- the lapsed time of the relay assertion in instructing the circuit breaker to open its contacts – traced by identifying the translated time sequence from rule 8 (t_8) to rule 15 (t_{15}) and eventually to rule 19(t_{19}), and
- the affected pole(s) – determined from the decision attribute *Trip* value.

Decision rules 2′, 3′, 4′, and 5′ are the consistent ones that constitute the positive region of the CD-decision algorithm.

The core $CORE(C,D) = \{bg, Z1pu\}$ can be justified why it is so. By dropping the attributes *bg* or *Z1pu*, one step at a time, their indispensability can be seen and whether the positive region that consists of the consistent rules changes can be checked. Different positive region is obtained by removing attribute *bg*:

rule 1′ (1, 2, 3, 20, 21): $Z1pu_0 \rightarrow Trip_0$

rule 2′ (4): $Z1pu_0 \rightarrow Trip_0$

rule 3′ (5, 6, 7): $Z1pu_0 \rightarrow Trip_0$

rule 4′ (8, 9, 10): $Z1pu_1 \rightarrow Trip_b$

rule 5′ (11, 12, 13, 14, 15): $Z1pu_1 \rightarrow Trip_b$

rule 6′ (16, 17, 18, 19): $Z1pu_0 \rightarrow Trip_b$

Likewise, the positive region can be changed as well by removing attribute Z1pu:

rule 1′ (1, 2, 3, 20, 21): $bg_0 \rightarrow Trip_0$

rule 2′ (4): $bg_1 \rightarrow Trip_0$

rule 3′ (5, 6, 7): $bg_2 \rightarrow Trip_0$

rule 4′ (8, 9, 10): $bg_1 \rightarrow Trip_b$

rule 5′ (11, 12, 13, 14, 15): $bg_0 \rightarrow Trip_b$

rule 6′ (16, 17, 18, 19): $bg_0 \rightarrow Trip_b$

Thus, when one by one the said condition attributes is removed, the changes incurred in the positive region of the relay *CD*-decision algorithm concur with the core attributes' indispensability. Thus, the core having both attributes {*bg, Z1pu*} is correct.

4.5. Protective relay decision algorithm minimization

It is subsequently desirable to further minimize the decision rules in the relay *CD*-decision algorithm after the above simplification via reduction of the set of condition attributes. This is achieved by removal of any possibly superfluous decision rules which essentially involves reducing the superfluous values of attributes. In other words, the unnecessary conditions have to be separately removed leaving only the core attribute in each decision rule of the algorithm [10].

The tabulated version of the above simplified relay *CD*-decision algorithm is shown in Table 10.

U	*bg*	*Z1pu*	*Trip*
1′ (1, 2, 3, 20, 21)	0	0	0
2′ (4)	1	0	0
3′ (5, 6, 7)	2	0	0
4′ (8, 9, 10)	1	1	a
5′ (11, 12, 13, 14, 15)	0	1	a
6′ (16,17,18,19)	0	0	a

Table 10. *DS* of simplified *CD*-decision algorithm

In Table 11 the condition attribute of each decision rule in Table 10 is removed one by one. In each removal the resultant rule is cross checked with other rules to find whether they are in conflict (inconsistent). This cross reference with other rules is to figure out whether the remaining condition attribute's value is the same but implication on the decision attribute is different. This process discovers the core attribute(s) that when eliminated causes the corresponding decision rule, or in general the *CD*-decision algorithm, inconsistent and consequently invalid (albeit not necessarily in the relay analysis perspective).

In summary, Table 12 contains cores of each decision rule. The condition attribute having eliminated value can be said as having no effect whatsoever on the *CD*-decision algorithm and may be termed as "don't care". It can be assigned with a value or otherwise. Combining duplicate rules and demarcating separate decision classes, Table 13 is obtained.

For decision attribute *Trip* = 0, one *minimal* set of decision rules is obtained from

$$bg_0\ Z1pu_0 \rightarrow Trip_0$$

$$bg_1\ Z1pu_0 \rightarrow Trip_0$$

$$bg_2 \rightarrow Trip_0$$

i.e.

$$bg_0\ Z1pu_0 \vee bg_1\ Z1pu_0 \vee bg_2 \quad \rightarrow Trip_0$$

For decision attribute $Trip = a$, one minimal set of decision rules is obtained from

$$Z1pu_1 \rightarrow Trip_b$$

$$bg_0\ Z1pu_0 \rightarrow Trip_b$$

i.e.

$$Z1pu_1 \vee bg_0\ Z1pu_0 \rightarrow Trip_b$$

The combined form of the *minimal CD*-decision algorithm is

$$bg_0\ Z1pu_0 \vee bg_1\ Z1pu_0 \vee bg_2 \quad \rightarrow Trip_0$$

or,

$$Z1pu_0\ (bg_0 \vee bg_1) \vee bg_2 \rightarrow Trip_0$$

and

$$bg_0\ Z1pu_0 \vee Z1pu_1 \rightarrow Trip_b$$

The final form of *CD*-decision algorithm can now be easily interpreted as follows:

- The decision rule $Z1pu_0\ (bg_0 \vee bg_1) \vee bg_2 \rightarrow Trip_0$ is interpreted as,

  IF $Z1pu = 0$ AND either $bg = 0$ OR $bg = 1$ OR IF $bg = 2$, THEN $Trip = 0$.

  The non-trip assertion ($Trip = 0$) is imminent with either one of the following situations:

  i. when no fault occurs ($bg = 0$) and no relay pick-up ($Z1pu = 0$), or
  ii. when a A-G fault occurs in zone 1($bg = 1$) and no relay pick-up ($Z1pu = 0$), or
  iii. when a A-G fault occurs in zone 2 ($bg = 2$)

- The decision rule $bg_0\ Z1pu_0 \vee Z1pu_1 \rightarrow Trip_b$ is interpreted as,

  IF $Z1pu = 0$ AND $bg = 0$ OR IF $Z1pu = 1$, THEN $Trip = b$.

  The trip assertion ($Trip = b$) is imminent with either one of the following situations:

  i. when there is no more fault indication ($bg = 0$) and relay pick-up element has reset ($Z1pu = 0$), or

  ii. when relay pick-up element remains asserted ($Z1pu = 1$)

  Item i. indicates the fact that trip assertion $Trip = b$ is still present in the face of the fault and relay pick-up resets (i.e. $bg = 0$ and $Z1pu = 0$) suggests that the preset time duration of the trip assertion is taking place.

CD-decision algorithm	Removed attribute *bg*	Removed attribute *Z1pu*	Resultant rule to check	At least one other rule in conflict	Core attribute
rule 1′ (1, 2, 3, 20, 21): $bg_0\ Z1pu_0 \rightarrow Trip_0$	✓		$Z1pu_0 \rightarrow Trip_0$	rule 6′: $Z1pu_0 \rightarrow Trip_b$	*bg*, *Z1pu*
		✓	$bg_0 \rightarrow Trip_0$	rule 5′: $bg_0 \rightarrow Trip_b$	
rule 2′ (4): $bg_1\ Z1pu_0 \rightarrow Trip_0$	✓		$Z1pu_0 \rightarrow Trip_0$	rule 6′: $Z1pu_0 \rightarrow Trip_b$	*bg*, *Z1pu*
		✓	$bg_1 \rightarrow Trip_0$	rule 4′: $bg_1 \rightarrow Trip_b$	
rule 3′ (5, 6, 7): $bg_2\ Z1pu_0 \rightarrow Trip_0$	✓		$Z1pu_0 \rightarrow Trip_0$	rule 6′: $Z1pu_0 \rightarrow Trip_b$	*bg*
		✓	$bg_2 \rightarrow Trip_0$	none	
rule 4′ (8, 9, 10): $bg_1\ Z1pu_1 \rightarrow Trip_a$	✓		$Z1pu_1 \rightarrow Trip_b$	none	*Z1pu*
		✓	$bg_1 \rightarrow Trip_b$	rule 2′: $bg_1 \rightarrow Trip_0$	
rule 5′ (11, 12, 13, 14, 15): $bg_0\ Z1pu_1 \rightarrow Trip_b$	✓		$Z1pu_1 \rightarrow Trip_b$	none	*Z1pu*
		✓	$bg_0 \rightarrow Trip_b$	rule 1′: $bg_0 \rightarrow Trip_0$	
rule 6′ (16, 17, 18, 19): $bg_0\ Z1pu_0 \rightarrow Trip_b$	✓		$Z1pu_0 \rightarrow Trip_b$	rule 1′: $Z1pu_0 \rightarrow Trip_0$	*bg*, *Z1pu*
		✓	$bg_0 \rightarrow Trip_b$	rule 1′: $bg_0 \rightarrow Trip_0$	

Table 11. Eliminating unnecessary condition attribute in decision rules

U	*bg*	*Z1pu*	*Trip*
1′	0	0	0
2′	1	0	0
3′	2	-	0
4′	-	1	b
5′	-	1	b
6′	0	0	b

Table 12. Cores of decision rules

U	*bg*	*Z1pu*	*Trip*
1′	0	0	0
2′	1	0	0
3′	2	-	0
4″ (4′,5′)	-	1	b
6′	0	0	b

Table 13. Cores of decision rules

5. Conclusion

Rough set theory has been proven to be an essentially useful mathematical tool in intelligent data mining analysis of inconsistent and vague protective relay data pattern as evident in the rough classification involved in the assertion of the trip decision attribute. The adoption of rough set theory is managed under supervised learning.

A single *D*-reduct of *C* (i.e. $RED_D(C) = \{bg, Z1pu\}$) has been discovered after formulating the attribute priority of the distance relay operation to trim the *DS*. $RED_D(C)$ can alternatively be used to represent exactly the same equivalence relation $U|IND(D)$ represented by the whole set of attributes *C*. Relying on the reduced number of condition attributes represented by $RED_D(C)$, relay analysis that can be achieved at ease.

The *D*-core of *C* (i.e. $CORE_D(C) = \{bg, Z1pu\}$), determined as the set of all single entries of the *D*-discernibility matrix, provides us with a novel technique in inferring the power system state where the relay has been subjected to. The core, because of its indispensability nature, draws our attention undoubtedly to the fact that an B-G fault has occurred and consequently the relay's Z1 ground distance element has picked up to eliminate it. This eventually translates into the trip decision having patterns such as that presented by the attribute *Trip*.

The degree of dependency $k < 1$ of the relay *CD*-decision algorithm justifies our anticipation of rough classification in the distance relay data. This is evidently shown in some of the rules that have the decision attribute *Trip* remain asserted with the value "b" for a certain preset duration of time. This is irrespective of the presence or absence of the fault via the assertion of attribute "*bg*" and zone 1 element pick-up via the assertion of attribute "*Z1pu*").

The $RED(C,D) = \{bg, Z1pu\}$ provides us with the discovery of the relay *CD*-decision algorithm in a simple form. By eliminating any possible superfluous decision rules, isolating condition attributes, one value at a time, further minimization of the algorithm can be performed.

Author details

Mohammad Lutfi Othman
Department of Electrical and Electronic Engineering, Faculty of Engineering, Universiti Putra Malaysia, Serdang, Malaysia

Ishak Aris
Department of Electrical and Electronic Engineering, Faculty of Engineering, Universiti Putra Malaysia, Serdang, Malaysia

Acknowledgement

This work was supported by the Ministry of Higher Education Malaysia under the 2011 Fundamental Research Grant Scheme with the project code FRGS/1/11/TK/UPM/03/4 and project file code 1057FR.

6. References

[1] Bakar A H A (2001) Disturbance Analysis in TNB Transmission System. Developments in Power System Protection Conference. IEE Publication No.479: 339-442
[2] Kumm J J, Weber M S, Schweitzer E O and Hou D (1994) Philosophies For Testing Protective Relays. 48th Annual Georgia Tech Protective Relaying Conference, Atlanta, Georgia.
[3] Kezunovic, M (2001) Section II: Equipment characteristics. IEEE Tutorial on Automated Fault Analysis. Texas A&M University, College Station, USA, July, pp 5-9.

[4] Kezunovic, M (2001) Section III: Scope of Analysis. IEEE Tutorial on Automated Fault Analysis. Texas A&M University, College Station, USA, July, pp 10-13.

[5] Shapiro G P (1997) Data Mining and Knowledge Discovery: The Third Generation, Foundations of Intelligent Systems. Berlin / Heidelberg: Springer. vol. 1325/1997, pp. 48-49.

[6] Kim G C, Mohd Zin A A and Shukri Z (2005) COMTRADE-Based Fault Information System for TNB Substations. IEEE Proc. Region 10 TENCON, Melbourne, Australia. pp. 1-6.

[7] Othman M L, Aris I, Abdullah S M, Ali M L and Othman M R (2009) Discovering Decision Algorithm from a Distance Relay Event Report. European j. scientific research, 3(1): 30-56.

[8] Pawlak Z (2002) Rough Set and Intelligent Data Analysis. Int. j. information sciences. 147: 1-12.

[9] Huang C L, Li T S and Peng T K (2005) A Hybrid Approach Of Rough Set Theory And Genetic Algorithm For Fault Diagnosis. Int. j. advanced manufacturing technology, Springer-Verlag London, 27(1-2): 119-127.

[10] Pawlak Z (1991) Rough Sets – Theoretical Aspects of Reasoning about Data. Dordrecht/Boston/London: Kluwer Academic Publishers.

[11] Walczak B and Massart D L (1999) Rough Sets Theory: Tutorial", Chemometrics and intelligent laboratory systems. 47(1): 1-16.

[12] Pawlak Z (1995) Rough Set Approach To Knowledge-Based Decision Support. 14th European j. operational research, Jerusalem, Israel, pp. 48–57.

[13] Hor C L and Crossley P A (2006) Substation Event Analysis Using Information from Intelligent Electronic Devices. Int. j. electrical power and energy systems, 28(6): 374-386.

[14] Pawlak Z, Wong S K M and Ziarko W (1988) Rough Sets: Probalistic versus Deterministic Approach. Int. j. man-machine studies. 29: 81-95.

[15] Komorowski J, Pawlak Z, Polkowski L and Skowron A (1999) Rough Fuzzy Hybridization—A New Trend in Decision Making. In: Pal S K, Skowron A, editors. Rough Sets: A Tutorial. New York: Springer.

[16] Ziarko W and Shan N (1995) Discovering Attribute Relationships, Dependencies And Rules By Using Rough Sets. Proceedings of the 28th Annual Hawaii International Conference on System Sciences (HICSS '95), pp. 293–299.

[17] Pawlak Z (2003) Rough Sets. Technical paper, Institute of Theoretical and Applied Informatics, Polish Academy of Sciences, Gliwice, and University of Information Technology and Management, Warsaw, Poland. pp. 1-51.

[18] Pawlak Z (2004) Some Issues on Rough Sets. In: Peters J F et al., editors. Transactions on Rough Sets I LNCS 3100. Verlag, Berlin, Heidelberg: Springer. pp. 1-58.

[19] Skowron A and Rauszer C (1992) The Discernibility Matrices and Functions in Information System. In: Slowinski R, editors. Intelligent Decision Support: Handbook of Applications and Advances of the Rough Sets Theory. Series D: System Theory, Knowledge Engineering and Problem Solving. Dordrecht: Kluwer Academic Publishers. 11, pp. 331-362.