symmetry

MDPI

Article

Using Rough Set Theory to Find Minimal Log with Rule Generation

Tahani Nawaf Alawneh * and Mehmet Ali Tut

Department of Mathematics, Faculty of Arts and Sciences, Eastern Mediterranean University, Via Mersin 10, Famagusta 99628, North Cyprus, Turkey; mehmet.tut@emu.edu.tr
* Correspondence: tahani.alawneh@emu.edu.tr; Tel.: +90-627-9739-0333

Abstract: Data pre-processing is a major difficulty in the knowledge discovery process, especially feature selection on a large amount of data. In literature, various approaches have been suggested to overcome this difficulty. Unlike most approaches, Rough Set Theory (RST) can discover data dependency and reduce the attributes without the need for further information. In RST, the discernibility matrix is the mathematical foundation for computing such reducts. Although it proved its efficiency in feature selection, unfortunately it is computationally expensive on high dimensional data. Algorithm complexity is related to the search of the minimal subset of attributes, which requires computing an exponential number of possible subsets. To overcome this limitation, many RST enhancements have been proposed. Contrary to recent methods, this paper implements RST concepts in an iterated manner using R language. First, the dataset was partitioned into a smaller number of subsets and each subset processed independently to generate its own minimal attribute set. Within the iterations, only minimal elements in the discernibility matrix were considered. Finally, the iterated outputs were compared, and those common among all reducts formed the minimal one (Core attributes). A comparison with another novel proposed algorithm using three benchmark datasets was performed. The proposed approach showed its efficiency in calculating the same minimal attribute sets with less execution time.

Keywords: rough set theory; R language; discernibility matrix

Citation: Alawneh, T.N.; Tut, M.A. Using Rough Set Theory to Find Minimal Log with Rule Generation. *Symmetry* **2021**, *13*, 1906. https://doi.org/10.3390/sym13101906

Academic Editors: Yagub Sharifov, Nazim Mahmudov and Peng-Yeng Yin

Received: 6 August 2021
Accepted: 5 October 2021
Published: 10 October 2021

1. Introduction

Information system security has been achieved using several security solutions such as IDS, IPS, anti-viruses and firewalls, etc.. Each device will work independently to guarantee appropriate access to network resources [1–4], and will generate its own alert logs, bearing in mind that these logs are growing too rapidly to be covered under the terminology big data. High repetitions and false alerts are common in such logs. As a result, this may mislead the process of identifying real threats. This poses a difficulty in analyzing large logs and detecting serious security issues and intrusions, as well as reacting at the right time [5,6].

To address these issues, several event correlation approaches have been proposed. Some approaches depend on similarity in records, others depend on machine learning and some depend on pattern recognition, etc. All approaches use predefined rules to gather and examine logs from different devices [7–9]. Fortunately, many of these approaches are implemented in real security systems, such as IBM SIEM, OSSIM, McAfee/Intel [10,11]. However, a commonality of all approaches is the large computation required due to multidimensional attributes of security logs, which prevents some of them from being employed in actual systems [11]. However, if a minimal log set can be created, under the condition of remaining consistent to decision attribute(s), the problem of repetition and real-time detection can be solved [12,13]. Luckily, RST can discover dependency within datasets and reduce the number of attributes using the data itself, without need for supplementary information.

The authors of [11] proposed a method to remove unnecessary attributes from network security logs using a rough set reduction algorithm. Moreover, the paper created a rule database using the algorithm of pattern mining, which depended on the timestamp of events. The work of this paper was very important for our research. It is similar to using rough set algorithms in general, but in our research, we performed attribute reduction to create a minimal optimal set.

The study performed by Y. Yao, et al. [12] was similar in the general use of the concepts of RST, but had a different purpose, as well as a different methodology. To identify security semantics, the researchers employed RST to analyze alert data collected from multiple sources. This was performed by collecting security data from several resources, then applying RST concepts. Weight was calculated for classifications of alerts, then alert aggregation was performed to eliminate repetitive and false alerts. Finally, a reliability metric was introduced according to background information to measure credibility. Therefore, our research would be a complement to this work as an extra stage to enhance both the classification of alerts and credibility measuring.

In ref. [14], the author explained the power of RST in knowledge discovery based on real world big datasets. These datasets can be uncertain, imprecise, and incomplete, which may misguide data analysis. Rough set feature selection algorithms can handle such problems by selecting the most relevant features to provide better results with less probability of information loss. The paper discussed the quickreduct algorithm, relative reduct algorithm, and entropy-based reduct algorithm. Lastly, output comparison was discussed.

In ref. [15], the authors proposed an effective and scalable rough set theory-based approach for pre-processing large scale datasets to perform feature selection. The idea of the approach was to use distributed parallel algorithms to enhance the execution time of data analysis. As a result, this made it possible to adapt the approach to pre-processing big data.

M.R. Gauthama Raman et al. [16] presented a selection method to determine the optimal attribute subset of the Intrusion Detection System (IDS). This technique used RST and some properties of the hypergraph to enhance the accuracy of classification and time complexity of IDS.

Dutta, S., Ghatak, S., Dey, R. et al. [17] proposed an attribute selection methodology that improved spam classification for Online Social Network (OSN). RST concepts were applied to develop an attribute selection algorithm to identify a smaller group of features that led to improving classification performance.

Anitha, A., and D. P. Acharjya [18] proposed a feature selection technique based on novel filter stands on the RST approach and Hyper-Clique based Binary Whale Optimization Algorithm (RST-HCBWoA). The technique identified informative features. This was necessary for an effective feature selection algorithm used in supervisory control and data acquisition IDS to protect critical infrastructure from cyber-attacks.

Nanda, N.B. and Parikh, A. [19] used RST to propose a hybrid technique that worked on identified risks of the network-attached intrusion detection system to determine the minimum rules set that could represent the knowledge offered by the data set under consideration. The two models used in this procedure were random forest classifier, to select attributes, and RST, to generate rules.

Hence, based on the work in [16–19], our study proposes more relevant research, providing a novel algorithm by using rough set package in R language to find the optimal minimal subset of attributes, rather than a smaller one without sacrificing performance.

In this paper, a new technique is proposed to create a minimal data log in addition to recent reduction approaches, particularly those mentioned in [20,21]. This minimal log will be used to create a minimal decision rule database. The motivation for proposing this methodology is to overcome the prohibitive complexity of RST concepts when searching for an optimal attribute subset, especially with big data. Offering such solutions will enhance the efficiency of real-time analysis of security algorithms, i.e., real-time IPS. The research contributions are:

- Developing a new algorithm using RST basic concepts to create minimal re-ducts;
- Offering a feasible feature selection methodology scalable to huge datasets, without sacrificing performance;
- Creating a minimal rule decision database that retains information content;
- Using three benchmark UCI datasets to evaluate the performance of the methodology;
- Comparing the result of the proposed model to recent works.

The rest of the paper is organized as follows: Section 2 discusses the related works. Section 3 gives a theoretical background about the rough set and R language. Section 4 discusses the methodology, motivation, datasets used, the proposed algorithm, and shows the experimental comparison results. Section 5 concludes the paper.

2. Related Works

As feature selection denotes the operation of selecting a subset of attributes from the original large set of attributes, the selected subset should be of the most important and relevant attributes among all. For example, in biomedical problems [22], it is the process of locating key genes, given a huge number of options; in business, it is the process to discover core key indicators for growth [23]; and in text-mining, it is the way to choose key terms [24].

For data mining and machine learning real-world issues, irrelevant misleading features, as well as noise data, occur. Therefore, the process of feature selection has gained high importance, since pre-processing the data can overcome such problems. To accomplish the task, two concepts are used: relevancy and redundancy of the feature. We say that a feature is relevant when it can predict decision feature(s), and that a feature is redundant when it offers the same value of information regarding any context, meaning that it had a high correlation. Hence, the successful feature selection process should detect attributes that have a high correlation with decision feature(s), however uncorrelated with themselves.

In consideration of the literature, it is worth mentioning that a comprehensive study was performed in [25], where the authors conducted a deep analysis regarding the scalability of existing recent techniques for feature selection. The paper concluded that recent methods would have scalability difficulties when processing big data. The authors showed that these difficulties would be in the process of handling large attributes, in both training time as well as efficiency in choosing relevant features. They recommended redesigning existing algorithms and their activation in distributed and parallel environments/frameworks. They stated that to overcome the limitations of recent approaches, a tool that requires no external/supplementary information is needed, and fortunately, RST can be used [26].

To overcome the weakness of RST in processing big data, several methods have been developed in the literature. The authors in [27] developed an evolutionary MapReduce algorithm, and used a parallel genetic algorithm to calculate the minimum rough set reduct. Within the different contexts in [28], the authors introduced a theoretical framework named local rough set. Here, a series of attributes reduction and approximation algorithms were developed with linear time complexity. However, it only fits limited labeled big data. The authors in [29] worked on developing a distributed definition of RST to treat reductions in information systems.

The authors of [30] produced a reduced maximal discernability pairs concept, based on fuzzy RST object pair framework. They developed two separate algorithms related to the concept. Results after comparison indicated that both algorithms were feasible and efficient.

As mentioned in [31], the authors based an RST proposed enhancement on two feature selection algorithms: Quickreduct, and entropy-based reduct, to find the minimal feature subset, because both algorithms have drawbacks in determining such minimal reduct. The measured performance using benchmark databases showed that the enhancement had overcome the drawbacks in terms of running time and minimal attribute calculation.

In [20], to overcome the limitations of the high computations of RST algorithms in features selection, when used to process large-scale data, the authors proposed a scalable

rough set theory-based approach for data pre-processing, particularly to select features using the Spark framework. The experimental results showed that the solution performed well in feature selection, which made it suitable to be used with big data.

The authors of [21] proposed a novel algorithm for finding optimal reduct using fuzzy RST. The idea was to consider only the minimal attributes in the discernibility matrix when performing the calculation of reducts. The experimental comparison proved the efficiency of the proposed algorithm. In Section 4.4 of this paper, the effectiveness of our proposed algorithm is proved by comparing its performance with the statistical results mentioned in [21].

3. Theoretical Background

3.1. Rough Set

Rough set theory (RST) is a mathematical approximation of standard set theory that allows for decision-making approximations [32,33]. This method can retrieve knowledge from a problem area in a succinct manner while retaining information contents and decreasing the amount of data involved [34]. The principles of RST to achieve feature selection are explained below.

Rough set training data is referred to as an information table or an information system. It is represented as a table, with rows representing instances or objects, and columns representing features or properties. A tuple can be used to represent the information table as S = (U, A).

$U = \{u_1, u_2, \ldots, u_n\}$ is called the universe set, which is a finite non-empty set of N objects (or instances), and A is (n + k) attribute set, which is non-empty. The set A (A = C ∪ D) is split into the following two subsets: conditional attribute set C and decision attribute D. The subset $C = \{a_1, a_2, \ldots, a_n\}$ has n predictors or conditional attributes, while the subset $D = \{d_1, d_2, \ldots, d_k\}$ has k output variables or decision attributes. For every single feature $a \in A$, there exists a domain which collects possible assigned values denoted by V_a.

A core notion of rough set theory is indiscernibility relation P, which is a binary relation defined as follows for every non-empty subset of attributes $P \subset C$:

$$IND(P) = \{(u_1, u_2) \in U \times U : \forall a \in P, a(u_1) = a(u_2)\}. \quad (1)$$

Here, $a(u_i)$ indicates the attribute value for the object i. This shows that if two objects belong to indiscernibility relation $(u_1, u_2) \in IND(P)$, then, by attributes P, u_1 is indistinguishable or unidentifiable (indiscernible) from u_2. The relation mathematically is symmetric, reflexive, and transitive. Now let $[u]_P$ be the set representing the generated equivalence classes, where $u \in U$. This set divides U into distinct classes or blocks labeled as U/P.

Any objects set taken from the universe set $X \subseteq U$ can be approximated by using equivalence classes produced by P as shown:

$$\underline{P}(X) = \{u : [u]_P \subseteq X\} \quad (2)$$

$$\overline{P}(X) = \{u : [u]_P \cap X \neq \varnothing\} \quad (3)$$

$\underline{P}(X)$ is called P-lower, which contains objects that definitely belong to X, while $\overline{P}(X)$ is called P-upper, which contains objects that possibly belongs to X. Both P-lower and P-upper are called approximations of the set X.

The difference between the two approximations is called the boundary region. It contains a set of objects that can possibly, but not certainly, be classified in a specific way. If this difference produces an empty set, this would be a precise or exact approximation, and we would say that X is actually crisp set $\overline{P}(X) = \underline{P}(X)$, or else, the set is rough.

Comparing attribute subsets is possible using a concept called dependency. For example, to measure the dependency of a subset of attributes Q, on a subset of attributes P, the following formula is used:

$$\gamma_P(Q) = |POS_P(Q)| / |U|, 0 \leq \gamma_P(Q) \leq 1, |\ | \text{ means cardinality} \quad (4)$$

$$POS_p(Q) = U_{x \in [u]Q} P(X) \quad (5)$$

where POS(Q) denotes Q positive region regarding P, it collects all objects of U which are distinctively categorized to classes of partition $[u]_Q$ using P. The closer γ_P (Q) is to 1, the more dependent Q is on P. RST proposes two essential ideas for feature selection based on these fundamentals, which are the Core, and the Reduct.

Rough set theory aims to generate a smaller subset of a given conditional attribute(s) data set, but the reduced subset should remain consistently related to the conditional attribute(s) [35,36]. A dataset is considered consistent if the corresponding decision attributes are similar for any objects set with equal feature values. The theory does this by defining the Reduct and Core notions.

Technically, in any information table, unnecessary attributes can be classified as either irrelevant or redundant. The goal is to create a heuristic that establishes a metric for determining feature necessity, but the process is not easy. A rough set that defines the strong and weak relevance of an attribute in terms of the likelihood of the desired concept occurrence, provides this attribute. The set that contains relevant attributes which are classified as strong, will form indispensable features, because their elimination from the information table is not possible without producing prediction accuracy loss. Hence, the importance value of every feature can be provided.

Conversely, in some instances, the collection of weak relevant characteristics might add to prediction accuracy. According to rough set concepts, strong relevant attributes are translated to Core concept, while the reduct concept mixes some weak relevant attributes with strong ones. For the set C, a subset R is considered as a reduct of C if:

$$\gamma_R(D) = \gamma_C(D), \text{ where } R \subseteq C \quad (6)$$

where there exists no $R' \subset R$, such that $\gamma_{R'}(D) = \gamma_R(D)$, if this condition is satisfied, the reduct is called the minimal reduct, where the features selected are the minimum that preserve the same value of dependency degree as the whole original feature set. However, we should remember that the definition allows the theory to generate a set of possible reuducts, $RED^F{}_C(D)$, and any of them are allowed to be used.

The intersection of all generated reducts will form the core attribute set:

$$CORE_C(D) = \cap\, RED^F{}_C(D) \quad (7)$$

Core features is the most essential subset, where any feature cannot be deleted without producing a collapse in the structure of the equivalence class, and according to rough set concepts, features of the core subset are indispensable.

The discernibility matrix notion *M*(A) is worth mentioning. For the information table S = (U, A), a discernibility matrix *M*(A) is a symmetric matrix with (n × n) dimension, and its elements c_{ij} can be defined by:

$$c_{ij} = \{a \in A : a(xi) \neq a(xj)\} \quad for\ i.j = 1, \ldots, n \quad (8)$$

This means that each c_{ij} contains attributes for which x_i and x_j are different. If this matrix is adapted with any decision table, the definition will be:

$$c_{ij} = \begin{cases} \{a \in A : a(xi) \neq a(xj)\} & if\ d(xi) \neq d(xj) \\ \varnothing & otherwise \end{cases} \quad (9)$$

The matrix in this case is called (decision-relative) discernibility matrix, and in RST this matrix is unique.

3.2. R Language

R is a computer language that is designed for data visualization and data analysis based on concepts approximation, and is used in data mining, statistics, machine learning,

and bioinformatics [37,38]. R was created by Robert Gentleman and Ross Ihaka at the University of Auckland [39] in 1997. Currently, R has more than 5000 packages existing in Comprehensive R Archive Network (CRAN) and the Bioconductor project repositories at http://cran.r-project.org/ and at http://www.bioconductor.org/ (accessed on 21 February 2021) [40].

The RoughSets package in R implements the theory of rough set (RST) and fuzzy rough set (FRST) to model and analyze data. The package contains both fundamental concepts (indiscernibility relation, lower/upper approximation, etc.) and the implementation of such concepts in many procedures (discretization, instance selection, feature selection, nearest neighbor-based classifiers, and rule reductions). These details explain the advantages of using the R language RoughSets package over other available rough set tools such as Rough Set Data Explorer (ROSE), Rough Set Exploration System (RSES) and its enhancement (ROSETTA), Waikato Environment for Knowledge Analysis (WEKA), or Rough Set Based Intelligent Data Analysis System (RIDAS) [41], because RoughSets enables researchers to examine both the theoretical concept and its implementation for academic targets and further research, while other tools facilitate researchers to apply the concepts of the rough set without concentrating on learning basic theoretical knowledge [40].

Following this review of RST and R language, we will discuss three concepts: Fuzzy RST, uncertainty in RST, and Sensitivity Analysis (SA). This will help readers understand why we are using RST rather than Fuzzy SRT in the current paper, how to overcome uncertainty in RST using the R language, and to open the door towards a new research concept, SA, and how it could enhance our work.

Fuzzy RST is the generalization of RST, where Dubois and Prade added the concept of membership degree in fuzzy sets, to RST. The main advantage of this mix is to deal with datasets that have real-valued attributes without the need to perform extra treatment for the data such as discretization. The uncertainty concept is accepted now [42]. In addition, since our dataset had no real-valued attributes, RST algorithms of RoughSets package in R language were used in this research.

RST is a technique for dealing with uncertainty issues. An essential question of the theory is how to assess the uncertainty of knowledge. However, existing uncertainty metrics may not correctly capture the degree of uncertainty. This is because existing accuracy models only focus on specific aspects linked to the target set, ignoring its significant effect on the model. It is also because no one provides a precise definition of the uncertainty of knowledge in the approximation space. As a result, evaluating the accuracy and logic of a knowledge uncertainty measure is challenging [43]. Luckily, RoughSets package in R language has a variety of methods to calculate uncertainty, and these methods are implemented under BC.LU.approximation.FRST. Furthermore, another facility is provided, named "custom", where end-users can generate their own approximations by coding functions to calculate lower and upper approximations. Hence, we can use many scholars' suggested formulas on how to measure uncertainty in information such as those mentioned in [43–45] by implementing their outcome equations under the "custom" facility in RoughSets package in R language.

Sensitivity Analysis (SA), which is a relatively new research area, is the study of how a system's outputs are related to, and are impacted by, its inputs [46]. SA has the benefit of being an essential component of mathematical modeling [47], but what makes SA valuable to this research is its ability to perform dimensionality reduction, decision support, and data worth assessing. Dimensionality reduction is used to discover uninfluential variables in the system which can be redundant, hence those variables can be adjusted or removed in later investigations [48]. Conversely, decision support is used to assess the sensitivity of the expected result to various decision alternatives, assumptions, restrictions, and/or uncertainties. What-if scenarios are used to study the effect of a change in input(s) on a decision output. Finally, data worth assessing is used to identify processes, factors, and scales that dominantly influence a system, and for which new data gathering decreases targeted uncertainty the most [49].

According to several current criteria, SA was recently considered a prerequisite for effective modeling practice [50]. SA aims to take advantage of the factor sparsity principle. This principle states that, often only a small group of elements in a system have a substantial influence on given system output. Fortunately, R language has a package, named Sensitivity, which contains a collection of functions for global sensitivity analysis, factor screening, and robustness analysis [51]. This package can help enhance the work in this paper in terms of the new research field SA, taking advantage of uncertainty quantifications and real life decision-making support of SA, whether the systems have continuous or discrete variables.

4. Research Methodology

In this section, we will present our proposed iterated rough set based algorithm, which we name IRS. IRS is proposed to be scalable for big data pre-processing for feature selection. The algorithm generates a minimal security log from any given big data set. The proposed steps are employed to accelerate the run time. Then, as a result of generating a minimal log, a minimal decision rules database is generated; this decision set maintains the data consistency embedded in the original dataset. In this section, we will also clarify our IRS algorithm as an efficient solution able to perform big data feature selection with less execution time. It will be compared with an existing novel algorithm using three benchmark datasets to prove its effectiveness. However, we will first explain the motivation for proposing IRS by discussing the computational complexity of the traditional rough set theory when working with high dimensional datasets.

4.1. Problem Statement and Motivation

Performing feature selection when using RST will force the theory to compute each possible attributes combination. The number of attribute subsets that maybe created using m attributes from a set of N attributes is $\binom{N}{m} = \frac{N!}{m!(N-m)!}$ [52]. Hence, the number of generated feature subsets as a total, is $\sum_{i=1}^{N} \binom{N}{i} = 2^N - 1$. For instance, if $N = 30$, there will be around one billion possible combinations. This prevents the use of RST with high dimensional datasets. Moreover, hardware limitations exist, and in particular, memory capacity will not be able to store and calculate a huge number of entities. The RAM will need to allocate the entire dataset, its computations, and results. For big data this can exceed the physical memory. Our proposed algorithm was motivated by all of these reasons.

4.2. Datasets

In our research, we used four datasets, three of which were benchmark datasets taken from UCI [44]. The purpose was to examine the proposed algorithm's effectiveness. This will be discussed in more detail in Section 4.4. The fourth dataset was used to execute the proposed algorithm on real-life huge datasets.

Our real-life huge datasets were collected from a government enterprise that uses IBM security Qradar. Qradar is a Security Information and Event Management (SIEM) solution that collects and analyzes log data from security systems [53]. Three datasets were taken from Qradar, each containing 63,000 objects (Instances) with 10 attributes of unprocessed security events. This enterprise considers the cloud technology in its structure and virtual machine concepts for more than 40 servers that have both Microsoft and Linux operating systems. Part of the servers provide about 100 online services for citizens.

Table 1 shows the general structure of each SIEM dataset. Every dataset has 10 attributes, A = {Event Name, Log Source, Event Count, Low-Level Category, Source IP, Source Port, Destination IP, Destination Port, User Name, Magnitude}. The first 9 attributes are the condition attributes (C), while the last one, Magnitude, is the decision attribute {D}. Magnitude indicates the importance of the offense, and has an integer value ranging from one to eight, being from least severe to most severe. Hence, each dataset forms a decision table, T = (U, A ∪ D). The data populated in table T contain no real-valued attributes,

meaning that concepts of RST can be applied directly, without the need to perform extra pre-processing steps such as discretization [54].

Table 1. Decision table.

Event Name	Log Source	Event Count	Low Level Category	Source IP	Source Port	Destination IP	Destination Port	User Name	Magnitude
Tear down UDP connection	ASA @ 172.17.0.1	1	Fire wall Session Closed	8.8.8.8	53	172.18.12.10	53,657	N/A	7
Deny protocol src	R	1	Fire wall Deny	172.20.12.142	56,511	172.217.23.174	443	N/A	8
Deny protocol src	ASA @ 172.17.0.1	1	Fire wall Deny	172.20.18.54	52,976	213.139.38.18	80	N/A	8
Deny protocol src	ASA @ 172.17.0.1	1	Fire wall Deny	172.20.15.71	53,722	52.114.75.79	443	N/A	8
Deny protocol src	ASA @ 172.17.0.1	1	Fire wall Deny	192.168.180.131	55,091	40.90.22.184	443	N/A	8
Built TCP connection	ASA @ 172.17.0.1	1	Fire wall Deny	172.18.12.19	59,201	163.172.21.225	443	N/A	8

4.3. Building a Minimal Log Size (Reduct)

Considering [21,30], both papers discussed the concept of using maximal or minimal pairs in discernibility matrix to overcome the complexity of feature selection. We will use this concept in our methodology inside iterations calculation for the same reason. Later, the results of [21] will be used to prove the efficiency of our algorithm.

To compute any minimal subset, two mathematical foundations are needed: discernibility matrix, and reduct. These two concepts were previously explained in Section 3.1, while Equations (6) and (8) summarize the concepts. It was noted in Section 4.1 that this process is computationally expensive. The proposed IRS algorithm aims to overcome this limitation and reduce the execution time of minimal log generation by redesigning the calculations using two concepts: iteration calculations and minimal elements in the discernibility matrix calculations.

The iteration step divides the big dataset into N subsets and calculates the iterated minimal reduct for each, where finally the intersection of all previously calculated iterated minimal reducts will generate the core minimal feature subset. The second step focuses on reducing the calculation complexity in each iteration by passing only the minimal element in a discernibility matrix to reduce calculations. Working in such design will contribute to solving the problem in the following way:

- Splitting the dataset into N subsets and performing the proposed algorithm on each subset will overcome hardware limitations, since fewer entries means less memory space to upload the data, perform computations, and store the results. Keeping the whole high dimensional dataset in memory and performing all the previous steps, is mostly impossible;
- Reducing the number of calculations, since passing only the minimal elements in the discernability matrix to reducts calculation will not cause the computation of each possible attribute combination, and hence the equation $\sum_{i=1}^{N} \binom{N}{i} = 2^N - 1$ is no longer valid. This will certainly reduce the execution time. The proposed code is given in Algorithm 1:

Table 2 shows the output after performing the algorithm using our datasets. The three datasets are labelled as S1, S2, and S3, respectively. It was found that the server cannot run the whole data set with 6300 objects and 10 attributes at once, so each dataset was split into three parts and processed on three iterations (M = 3, N = 3). The table also shows the calculated degree of dependency for each iterated reduct, being how much the generated iterated reduct (attributes set) depends on the decision attribute(s), with a maximum value of 1. The methodology was performed under hardware specifications of Intel(R) Xeon(R) Gold 6148 CPU @2.40 GHz 2.39 GHz, RAM 48.3 GB.

Algorithm 1: IRS Algorithm

Input: T = (U,A∪D): information table, N: number of iterations,
M: number of datasets
Output: Core–Reduct,
1: **For** each dataset M do
2: **For** each iteration N do
3: Calculate $IND_N(D)$
4: Compute $DISC.Matrix_N(T)$
5: **Do while** ($DISC.Matrix_N(T) \neq \emptyset$) and $i \leq j$
(RST discernibility matrix is symetric)
6: $S_{i0,j0}$ = Sort $(x_i,x_j) \in DISC.Matrix_N(T)$
according to number of conditional attributes A
7: End while
8: Compute $Reduct_N(S_{i0,j0})$
(calculating reducts for minimal condition atrridutes)
9: $Reduct_N = Reduct_N \cap Reduct_N(S_{i0,j0})$
10: **End For** N
11: Core–Reduct = Core–Reduct ∩ $Reduct_N$
minimal optimal reduct
12: **End For** M

Table 2. Minimal reduct output for N = 3, M = 3.

Training Data Set	Minimal Attribute	Degree of Dependency [1]
First Training Set S1 (∩ three iterations) $Reduct_{N=1}$	A1 = {Event Name, Source IP, Source Port, Destination IP, Magnitude } | A1 | = 5	1
Second Training Set S2 (∩ three iterations) $Reduct_{N=2}$	A2 = { Event Name, Source IP, Destination IP, Magnitude } | A2 | = 4	0.9992941
Third Training Set S3 (∩ three iterations) $Reduct_{N=3}$	A3 = {Event Name, Source IP, Source Port, Destination IP, Magnitude } | A3 | = 5	1
Core-Reduct (A1∩ A2∩ A3)	A2 = { Event Name, Source IP, Destination IP, Magnitude } | A2 | = 4	0.9992941

[1]: a decision attribute, d, totally depends on a set of attributes A, written as A ⇒ d if all attribute values from d are distinctly identified by attribute values from A.

The intersection of the three iterations of the first data set S1 produced a minimal iterated reduct of 5 attributes \| ReductN = 1 \| = 5, while the original set S1 had 10 attributes. In this reduct, the degree of the dependency = 1, which means the decision attribute {Magnitude} was completely identified by the values of the 5 attributes of the reduct set A1. This reduct omitted 50% of the attributes of the original set and retained information content 100%. For the second iteration S2, the reduct was even better, producing 4 attributes, \| ReductN = 2 \| = 4, with a dependency degree of 0.9992941, while the last iteration for S3 had the same output as S1.

Following step 11 in the algorithm, Core-Reduct was generated by taking the intersection of all previous iterations' outputs. This means that Core-Reduct = {Event Name, Source IP, Destination IP, Magnitude} was the minimal reduct for all datasets S1, S2, and S3. It had 4 attributes, rather than the 10 attributes of the original sets. Despite this reduct, the information content of the original datasets was retained with 99.9% accuracy.

This proves that the proposed solution was able to create a minimal reduct for the security log, this optimal reduct used only 40% of the attributes of the original dataset (4 instead of 10), and still offered the same information covered by the original dataset with 99.9% accuracy degree. The next section will use this minimal dataset to create a minimal decision rules database. The effectiveness of step 8, which passes only the minimal elements of discernibility matrix for reduct calculations inside each iteration, will be proved in Section 4.4, by comparing our algorithm with a similar approach using the same benchmark datasets in terms of runtime.

4.4. Generating Minimal Decision Rules

It is significant to understand that the derivation of rule structure, using learning procedures from training cases, is being employed in rule-based expert systems. Fortunately, these rules are more accurate than information included in the original input data

set, because new examples that do not match examples taken from the original data, are being properly classified by such rules [34].

The RoughSets package in the R language has different algorithms to extract knowledge hidden in any given data set in the form of an IF . . . THEN structure. This paper uses the CN2Rules algorithm, which is designed to work even with the existence of imperfect data. The CN2Rules algorithm was deployed on each of the reduct sets A1, A2, and A3, produced in the previous section. The algorithm (Algorithm 2) below generates the decision rules in the form of an IF . . . THEN structure, with each set divided into a training set with 60%, and a test set with 40%, as shown in step 3, because this will be used later in steps 5 and 6 to validate the accuracy of the prediction using the 40% test part.

Algorithm 2: Rule Generation Algorithm

Input: Reduct_N (T): minimal reduct information table, M: number of datasets
Output: Set-Rule_{Min}
1: **For** each dataset M do
2: read.table(Reduct_N (T))
3: Splitting Reduct_N (T)
training set 60% and a test set 40%.
4: RI.LEM2Rules.RST() function
Create rules depending on training set of Reduct_N (T)
5: predict() function
Testing the quality of prediction depending on the test set of Reduct_N (T)
6: mean() function.
Checking the accuracy of predictions
7: **End For** M

Table 3 shows the total number of rules generated for each dataset before minimizing (S1, S2, S3), and after minimizing (A1, A2, A3). It also calculates the prediction accuracy of each minimal iterated reduct set (A1, A2, A3).

Table 3. Decision rules induction.

Training Data Set	Number of Decision Rules before Reduct	Number of Deccision Rules after Reduct	Prediction Accuracy
First Training Set	S1 = 905	A1 = 596	0.9552733
Second Training Set	S2 = 878	A2 = 509	0.9535073
Third Training Set	S3 = 813	A3 = 481	0.9741291

Examining the first row in the table and comparing the number of rules generated from the first original dataset S1 and its minimal reduct A1, the number of rules decreased to about 66% with a prediction accuracy of around 96%. A similar result occurred for datasets S1 and S2.

A minimal decision rules dataset was successfully created for each original dataset (S1, S2, S3). Each minimal decision rules dataset (A1, A2, A3) reduced the number of the rules (by 50% to 65%) with high accuracy prediction (from 95% to 97%). In addition, we know from the previous section that each minimal iterated set (A1, A2, A3) strongly represents the knowledge in its original dataset (S1, S2, S3) with a high degree of dependency (ranging from 1 to 0.99). We conclude that the same knowledge is being presented in the form of decision rules, with a smaller number of attributes and high accuracy prediction.

4.5. Execution Time Comparison with Existing Methods

The current section evaluates the efficiency of our IRS algorithm. Our technique for generating a minimal subset will be compared with two other techniques: one using classical discernibility matrix [55], while the other uses its own proposed novel algorithm, named

Sample pair selection SPS [21]. The experiments used the same hardware environment specifications mentioned in [21], being Intet (R) i5 CPU 2.40 GHz M450.

The comparison will measure the runtime needed to calculate the minimal reduct, using the three algorithms on the same datasets. It is worth noting that the comparison uses three benchmark datasets taken from the UCI machine learning repository [56]. Table 4 shows the description of the original dataset. These three datasets were previously used in [21,55] to compare the effectiveness of the SPS algorithm against the classical discernibility matrix.

Table 4. Original Datasets description.

Dataset	Number of Attributes	Number of Instances
Glass	9	100
Wiscon	9	699
Zoo	16	100

We executed our algorithm IRS using the three benchmark datasets and compared the runtime values with the previous statistical calculations from [21] and [55]. As shown in Table 5, our algorithm IRS generated the same number of all possible reducts for glass and Wiscon datasets 2 and 4 respectively. However, in the case of the zoo dataset, our algorithm created 35 reducts, while both compared algorithms created 33, yet our algorithm had the best runtime over the other two algorithms, at 0.9967 s. A general comparison of the runtime of all algorithms shows that our IRS algorithm had the best execution time over SPS and classical discernibility matrix, for all datasets. This proves that the IRS algorithm, which uses iteration calculations depending on minimal elements of discernibility matrix, decreased the complexity of calculations successfully.

Table 5. Computations of execution time in finding minimal reduct.

Data	Num. of Attributes of the Dataset	All Reducts		Execution Time in Seconds		
		IRS	SPS and CDM	Classical DiscernibilityMatrix (CDM)	SPS	IRS
Wiscon	9	4	4	1362.1	24.0956	9.05
Glass	9	2	2	23.3268	0.7931	0.7
Zoo	16	35	35	106.6581	1.2574	0.9967

5. Conclusions and Future Works

Following the proposed procedure, this paper designed a new algorithm named IRS to create a minimal security log. The approach used RST basic concepts by adopting an iterated model. Inside each iteration, minimal discernability matrix elements were passed for reduct calculations. This design helped to overcome hardware limitations and prevent reduct calculation growing exponentially high, by decreasing the calculation needed to compute all possible attribute combinations to the minimal elements in a discernibility matrix.

We also computed a minimal decision rule database with a prediction accuracy of about 96%. This minimal subset used only 40% of the attributes of the original feature set, with a 99.9% degree of dependency (knowledge consistency).

We compared our methodology with another recent novel algorithm, using the same three benchmark datasets. Our comparison showed that the proposed methodology effectively calculated a minimal set without losing performance. The results showed that our methodology was even better in terms of execution time, which proved that calculation complexity, as well as search space, were reduced. This makes the proposed model relevant to huge datasets, and will enhance real-time analyses.

In the future, we will apply the new concept of Sensitivity Analysis (SA) to this work, because SA can manage uncertainty in a real-world decision system, especially in high-dimensional problems. This will surely offer a better solution for work in this field of research.

Author Contributions: Data collection, data organization, simulation, result collection, conclusions, and future works, T.N.A.; problem definition and result commentary, M.A.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data set (in Table 2) used to support the findings of this study were supplied by security appliances owned by Aqaba Special Economic Zone Authority (ASEZA) and therefore cannot be made available. Requests for access to these data for reviewers should be made to Tahani Alawneh, tahani.alawneh@emu.edu.tr.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lundgren, B.; Moller, N. Defining information security. *Sci. Eng. Ethics* **2019**, *25*, 419–441. [CrossRef]
2. Bass, T. Intrusion detection systems and multisensor datafusion. *Commun. ACM* **2000**, *43*, 99–105. [CrossRef]
3. Xi, R.-R.; Yun, X.-C.; Jin, S.-y.; Zhang, Y.-Z. Research survey of network security situation awareness. *J. Comput. Appl.* **2012**, *32*, 1–4.
4. Lai, J.-B.; Wang, Y.; Wang, H.-Q.; Zheng, F.-B.; Zhou, B. Research on network security situation awareness system architecture based on multi-source heterogeneous sensors. *Comput. Sci.* **2011**, *38*, 144.
5. Yen, S.; Moh, M. Intelligent Log Analysis Using Machine and Deep Learning. In *Research Anthology on Artificial Intelligence Applications in Security*; IGI Global: Hershey, PA, USA, 2021; pp. 1154–1182.
6. Svacina, J.; Raffety, J.; Woodahl, C.; Stone, B.; Cerny, T.; Bures, M.; Shin, D.; Frajtak, K.; Tisnovsky, P. On Vulnerability and Security Log Analysis: A Systematic Literature Review on Recent Trends. In Proceedings of the International Conference on Research in Adaptive and Convergent Systems, Gwangju, Korea, 13–16 October 2020; Association for Computing Machinery: New York, NY, USA; pp. 175–180.
7. Chuvakin, A. Security event analysis through correlation. *Inf. Secur. J. Glob. Perspect.* **2004**, *13*, 13–18. [CrossRef]
8. Klemettinen, M.; Mannila, H.; Toivonen, H. Rule discovery in telecommunication alarm data. *J. Netw. Syst. Manag.* **1999**, *7*, 395–423. [CrossRef]
9. Bao, X.-H.; Dai, Y.-X.; Feng, P.-H.; Zhu, P.-F.; Wei, J. A detection and forecast algorithm for multi-step attack based on intrusion intention. *J. Softw.* **2005**, *16*, 2132–2138. [CrossRef]
10. Gonzalez-Granadillo, G.; Gonzalez-Zarzosa, S.G.; Diaz, R. Security information and event management(siem): Analysis, trends, and usage in critical infrastructures. *Sensors* **2021**, *21*, 4759. [CrossRef] [PubMed]
11. Liu, J.; Gu, L.; Xu, G.; Niu, X. A Correlation Analysis Method of Network Security Events Based on Rough Set Theory. In Proceedings of the 3rd IEEE International Conference on Network Infrastructure and Digital Content, Beijing, China, 21–23 September 2012; Institute of Electrical and Electronics Engineers: Piscataway, NJ, USA, 2012; pp. 517–520.
12. Yao, Y.; Wang, Z.; Gan, C.; Kang, Q.; Liu, X.; Xia, Y.; Zhang, L. Multi-source alert data understanding for security semantic discovery based on rough set theory. *Neurocomputing* **2016**, *208*, 39–45. [CrossRef]
13. Bao, L.; Li, Q.; Lu, P.; Lu, J.; Ruan, T.; Zhang, K. Execution anomaly detection in large-scale systems through console log analysis. *J. Syst. Softw.* **2018**, *143*, 172–186. [CrossRef]
14. Bania, R. Comparative review on classical rough set theory based feature selection methods. *Int. J. Comput. Appl.* **2015**, *114*, 31–35.
15. Dagdia, Z.C.; Zarges, C.; Beck, G.; Lebbah, M. A scalable and effective rough set theory-based approach for big data pre-processing. *Knowl. Inf. Syst.* **2020**, *62*, 3321–3386.
16. Raman, M.G.; Kirthivasan, K.; Sriram, V.S. Development of rough set–hypergraph technique for key feature identification in intrusion detection systems. *Comput. Electr. Eng.* **2017**, *59*, 189–200. [CrossRef]
17. Dutta, S.; Ghatak, S.; Dey, R.; Das, A.K.; Ghosh, S. Attribute selection for improving spam classification in online social networks: A rough set theory-based approach. *Soc. Netw. Anal. Min.* **2018**, *8*, 1–16. [CrossRef]
18. Anitha, A.; Acharjya, D. Crop suitability prediction in vellore district using rough set on fuzzy approximation space and neural network. *Neural Comput. Appl.* **2018**, *30*, 3633–3650. [CrossRef]
19. Nanda, N.B.; Parikh, A. Hybrid Approach for Network Intrusion Detection System Using Random Forest Classifier and Rough Set Theory for Rules Generation. In Proceedings of the 3rd International Conference on Advanced Informatics for Computing Research, Shimla, India, 15–16 June 2019; Springer: Cham, Switzerland, 2019; pp. 274–287.
20. Dagdia, Z.C.; Zarges, C.; Beck, G.; Lebbah, M. A distributed rough set theory based algorithm for an efficient big data pre-processing under the spark framework. In Proceedings of the 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, USA, 11–14 December 2017; pp. 911–916.

21. Chen, D.; Zhang, L.; Zhao, S.; Hu, Q.; Zhu, P. A novel algorithm for finding reducts with fuzzy rough sets. *IEEE Trans. Fuzzy Syst.* **2011**, *20*, 385–389. [CrossRef]
22. Ahmed, S.; Zhang, M.; Peng, L. Enhanced Feature Selection for Biomarker Discovery in LC-MS Data Using GP. In Proceedings of the 2013 IEEE Congress on Evolutionary Computation, Cancun, Mexico, 20–23 June 2013; Institute of Electrical and Electronics Engineers: Piscataway, NJ, USA, 2013; pp. 584–591.
23. Liu, H.; Yu, L. Toward integrating feature selection algorithms for classification and clustering. *IEEE Trans. Knowl. Data Eng.* **2005**, *17*, 491–502.
24. Aghdam, M.H.; Ghasem-Aghaee, N.; Basiri, M.E. Text feature selection using ant colony optimization. *Expert Syst. Appl.* **2009**, *36*, 6843–6853. [CrossRef]
25. Bolón-Canedo, V.; Rego-Fernández, D.; Peteiro-Barral, D.; Alonso-Betanzos, A.; Guijarro-Berdiñas, B.; Sánchez-Maroño, N. On the scalability of feature selec-tion methods on high-dimensional data. *Knowl. Inf. Syst.* **2018**, *56*, 395–442. [CrossRef]
26. Thangavel, K.; Pethalakshmi, A. Dimensionality re-duction based on rough set theory: A review. *Appl. Soft Comput.* **2009**, *9*, 1–12. [CrossRef]
27. El-Alfy, E.-S.M.; Alshammari, M.A. Towards scalable rough set based attribute subset selection for intrusion detection using parallel genetic algorithm in mapreduce. *Simul. Model. Pract. Theory* **2016**, *64*, 18–29. [CrossRef]
28. Qian, Y.; Liang, X.; Wang, Q.; Liang, J.; Liu, B.; Skowron, A.; Yao, Y.; Ma, J.; Dang, C. Local rough set: A solution to rough data analysis in big data. *International J. Approx. Reason.* **2018**, *97*, 38–63. [CrossRef]
29. Hu, J.; Pedrycz, W.; Wang, G.; Wang, K. Rough sets in distributed decision information systems. *Knowl. Based Syst.* **2016**, *94*, 13–22. [CrossRef]
30. Dai, J.; Hu, H.; Wu, W.-Z.; Qian, Y.; Huang, D. Maximal-discernibility-pair-based approach to attribute reduction in fuzzy rough sets. *IEEE Trans. Fuzzy Syst.* **2017**, *26*, 2174–2187. [CrossRef]
31. Velayutham, C.; Thangavel, K. Improved rough set algorithms for optimal attribute reduct. *J. Electron. Sci. Technol.* **2011**, *9*, 108–117.
32. Pawlak, Z. *Rough Sets: Theoretical Aspects of Reasoning about Data*; Springer Science & Business Media: Dordrecht, The Netherlands, 2012; Volume 9.
33. Pawlak, Z.; Skowron, A. Rudiments of rough sets. *Inf. Sci.* **2007**, *177*, 3–27. [CrossRef]
34. Peralta, D.; del Rio, S.; Ramirez-Gallego, S.; Triguero, I.; Benitez, J.M.; Herrera, F. Evolutionary feature selection for big data classification: A mapreduce approach. *Math. Probl. Eng.* **2015**, *2015*, 246139. [CrossRef]
35. Zbigniew, S. An Introduction to Rough Set Theory and Its Applications—A Tutorial. In Proceedings of the 1st International Computer Engineering Conference ICENCO'2004, Cairo, Egypt, 27–30 December 2004.
36. Ray, K.S. *Soft Computing and Its Applications: A Unified Engineering Concept*; CRC Press: Boca Raton, FL, USA, 2014; Volume 1.
37. Hothorn, T. *Cran Task View: Machine Learning & Statistical Learning*; The R Foundation: Vienna, Austria, 2021.
38. Rhys, H.I. *Machine Learning with R, the Tidyverse, and Mlr*; Manning Publications: Shelter Island, NY, USA, 2020.
39. Tuffery, S. *Data Mining and Statistics for Decision Making*; John Wiley & Sons: Hoboken, NJ, USA, 2011.
40. Aphalo, P.J. *Learn R: As a Language*; CRC Press: Boca Raton, FL, USA, 2020.
41. Abbas, Z.; Burney, A. A survey of software packages used for rough set analysis. *J. Comput. Commun.* **2016**, *4*, 10–18. [CrossRef]
42. Dubois, D.; Prade, H. Rough fuzzy sets and fuzzy rough sets. *Int. J. Gen. Syst.* **1990**, *17*, 191–209. [CrossRef]
43. Tang, J.; Wang, J.; Wu, C.; Ou, G. On uncertainty measure issues in rough set theory. *IEEE Access* **2020**, *8*, 91089–91102. [CrossRef]
44. Beaubouef, T.; Petry, F.; Arora, G. Information-theoretic measures of uncertainty for rough sets and rough relational databases. *Inf. Sci.* **1998**, *109*, 185–195. [CrossRef]
45. Parthaláin, N.M.; Shen, Q.; Jensen, R. A distance measure approach to exploring the rough set boundary region for attribute reduction. *IEEE Trans. Knowl. Data Eng.* **2010**, *22*, 305–317. [CrossRef]
46. Razavi, S.; Jakeman, A.; Saltelli, A.; Prieur, C.; Iooss, B.; Borgonovo, E.; Plischke, E.; Piano, S.L.; Iwanaga, T.; Becker, W.; et al. The future of sensitivity analysis: An essential discipline for systems modeling and policy support. *Environ. Model. Softw.* **2021**, *137*, 104954. [CrossRef]
47. Rodriguez, J.D.; Perez, A.; Lozano, J.A. Sensitivity analysis of k-fold cross validation in prediction error estimation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2009**, *32*, 569–575. [CrossRef] [PubMed]
48. Sobol, I.M.; Tarantola, S.; Gatelli, D.; Kucherenko, S.; Mauntz, W. Estimating the approximation error when fixing unessential factors in global sensitivity analysis. *Reliab. Eng. Syst. Saf.* **2007**, *92*, 957–960. [CrossRef]
49. Guillaume, J.H.; Jakeman, J.D.; Marsili-Libelli, S.; Asher, M.; Brunner, P.; Croke, B.; Hill, M.C.; Jakeman, A.J.; Keesman, K.J.; Razavi, S.; et al. Introductory overview of identifiability analysis: A guide to evaluating whether you have the right type of data for your modeling purpose. *Environ. Model. Softw.* **2019**, *119*, 418–432. [CrossRef]
50. Saltelli, A.; Bammer, G.; Bruno, I.; Charters, E.; di Fiore, M.; Didier, E.; Espeland, W.N.; Kay, J.; Piano, S.L.; Mayo, D.; et al. Five ways to ensure that models serve society: A manifesto. *Nature* **2020**, *582*, 482–484. [CrossRef]
51. Iooss, B.; Janon, A.; Pujol, G. *Sensitivity: Global Sensitivity Analysis of Model Outputs*; R Package Version 1.22.0; The R Foundation: Vienna, Austria, 2018.
52. Guyon, I.; Elisseeff, A. An introduction to variable and feature selection. *J. Mach. Learn. Res.* **2003**, *3*, 1157–1182.
53. Majeed, A.; Ur Rasool, R.; Ahmad, F.; Alam, M.; Javaid, N. Near-miss situation based visual analysis of siem rules for real time network security monitoring. *J. Ambient Intell. Humaniz. Comput.* **2019**, *10*, 1509–1526. [CrossRef]

54. Riza, L.S.; Janusz, A.; Bergmeir, C.; Cornelis, C.; Herrera, F.; Śle˛zak, D.; Benítez, J.M. Implementing algorithms of rough set theory and fuzzy rough set theory in the R package "roughsets". *Inf. Sci.* **2014**, *287*, 68–89. [CrossRef]
55. Tsang, E.C.C.; Chen, D.G.; Yeung, D.S.; Wang, X.Z.; Lee, J.W.T. Attributes reduction using fuzzy rough sets. *IEEE Trans. Fuzzy Syst.* **2008**, *16*, 1130–1141. [CrossRef]
56. UCI Machine Learning Repository. 2005. Available online: http://www.ics.uci.edu/mlearn/MLRepository.html (accessed on 3 September 2021).