Analysis of KDD '99 Intrusion Detection Dataset for Selection of Relevance Features

Adetunmbi A.Olusola., Adeola S.Oladele. and Daramola O.Abosede

Abstract - The rapid development of business and other transaction systems over the Internet makes computer security a critical issue. In recent times, data mining and machine learning have been subjected to extensive research in intrusion detection with emphasis on improving the accuracy of detection classifier. But selecting important features from input data lead to a simplification of the problem, faster and more accurate detection rates. In this paper, we presented the relevance of each feature in KDD '99 intrusion detection dataset to the detection of each class. Rough set degree of dependency and dependency ratio of each class were employed to determine the most discriminating features for each class. Empirical results show that seven features were not relevant in the detection of any class.

Keywords: Intrusion detection, machine learning, relevance feature, rough set, degree of dependency.

I. INTRODUCTION

As Internet keeps growing with an exponential pace, so also is cyber attacks by crackers exploiting flaws in Internet protocols, operating system and application software. Several protective measures such as firewall have been put in place to check the activities of intruders which could not guarantee the full protection of the system. Hence, the need for a more dynamic mechanism like intrusion detection system (IDS) as a second line of defense. Intrusion detection is the process of monitoring events occurring in a computer system or network and analyzing them for signs of intrusions [1]. IDSs are simply classified as host-based or network-based. The former operates on information collected from within an individual computer system and the latter collect raw networks packets as the data source from the network and analyze for signs of intrusions. The two different detection techniques employed in IDS to search for attack patterns are Misuse and Anomaly. Misuse detection systems find known attack signatures in the monitored resources. Anomaly detection systems find attacks by detecting changes in the pattern of utilization or bahaviour of the system.

Majority of the IDS currently in use are either rule-based or expert-system based. Their strengths depend largely on the ability of the security personnel that develops them. The former can only detect known attack types and the latter is prone to generation of false positive alarms. This leads to the use of an intelligence technique known as data mining/machine learning technique as an alternative to expensive and strenuous human input. These techniques automatically learn from data or extract useful pattern from data as a reference for normal/attack traffic behaviour profile from existing data for subsequent classification of network traffic.

Intelligent approach was first implemented in mining audit data for automated models for intrusion detection (MADAMID) using association rule [2]. Several others machine-learning paradigms investigated for the design of IDS include: neural networks learn relationship between given input and output vectors to generalize them to extract new relationship between input and output [3,4,5], fuzzy generalize relationship between input and output vector based on degree of membership [5,6], decision tree learns knowledge from a fixed collection of properties or attributes in a top down strategy from root node to leave node [5,7,8], support vector machine simply creates Maximum-margin hyper planes during training with samples from two classes [3,9,10] .

Rough sets produce a set of compact rules made up of relevant features only suitable for misuse and anomalous detection (9,11,12,13,14]. Bayesian approaches are powerful tools for decision and reasoning under uncertain conditions employing probabilistic concept representations [15,16].

Prior to the use of machine learning algorithms raw network traffic must first be summarized into connection records containing a number of within-connection features such as service, duration, and so on. Identification of important features is one of the major factors determining the success of any learning algorithm on a given task. Feature selection in learning process leads to reduction in computational cost, over fitting, model size and leads to increase in accuracy.

This manuscript was submitted May, 2007. The work was self sponsored.

Adeola, S. Oladele is an Oracle Certified Professional with a core competency is Oracle Database, Microsoft Basic, VB.Net, PL/SQL and Computer Networking. He is a member of professional bodies such as Computer Professional of Nigeria (CPN), Nigeria Computer Society (NCS), IEEE Computer society as well as Association of Computer Machinery (ACM). He has worked in different companies as a Network Engineer and Programmer. He was a consultant to a number of establishments in Nigeria including ALCATEL Nigeria and Nigeria Police Force. He is currently with the Federal University of Technology, P.M.B 704, Akure, Nigeria (phone: +234-8033749944; e-mail: deleadeola@yahoo.com)

Adetunmbi A. Olusola holds a PhD in Computer Science from the Federal University of Tech., Akure, Nigeria. He worked in different organization in Nigeria including Associated Business Information and Computer Services, Lagos, Nigeria. He was also a lecturer at Adeyemi College of Education, Ondo, Nigeria and University of Ado-Ekiti, Nigeria. He is a member of professional bodies such as Nigeria Computer Society, IEEE Computer Society and International Studies on Advanced Intelligence. He is currently a researcher with the Federal University of Tech., P.M.B 704, Akure, Nigeria (e-mail: bayo_adetunmbi@yahoo.com)

Daramola, O. Abosede holds a M.Tech degree in Computer science. He is a member of different professional bodies such as Nigeria Computer Society, Third World Organization of Women Scientists and Science Association of Nigeria. She is currently pursuing her PhD at the Department of Computer Science of the Federal University of Technology, Akure, P.M.B 704, Akure, Nigeria

Previous works in feature selection for intrusion detection include the work of [17, 18]. In this paper, attempt was made to investigate the relevance of each feature in KDD 99 intrusion detection dataset to substantiate the performance of machine learning and degree of dependency is used to determine the most discriminating features for each class. Therefore, the relevance of the forty one (41) features with respect to dataset labels was investigated.

This paper is organized as follows: in section 2 description of the intrusion detection evaluation dataset is presented followed by brief description of rough set and discretization technique employed in section 3. Section 4 presents the experimental setup and results followed by conclusion in section 5.

II. INTRUSION DETECTION DATASET

The KDD Cup 1999 dataset [21] used for benchmarking intrusion detection problems is used in our experiment. The dataset was a collection of simulated raw TCP dump data over a period of nine weeks on a local area Network. The training data was processed to about five million connections records from seven weeks of network traffic and two weeks of testing data yielded around two million connection records. The training data is made up of 22 different attacks out of the 39 present in the test data. The known attack types are those present in the training dataset while the novel attacks are the additional attacks in the test datasets not available in the training data sets. The attacks types are grouped into four categories:

(1).DOS: Denial of service – e.g. syn flooding
(2).Probing: Surveillance and other probing, e.g. port scanning
(3).U2R: unauthorized access to local super user (root) privileges, e.g. buffer overflow attacks.
(4).R2L: unauthorized access from a remote machine, e.g. password guessing

The training dataset consisted of 494,021 records among which 97,277 (19.69%) were normal, 391,458 (79.24%) DOS, 4,107 (0.83%) Probe, 1,126 (0.23%) R2L and 52 (0.01%) U2R connections. In each connection are 41 attributes describing different features of the connection and a label assigned to each either as an attack type or as normal. Table 1 shows the class labels and the number of samples that appears in “10% KDD” training dataset. Appendix II gives the detail description of KDD 99 Intrusion Detection Dataset Features.

II. BASIC CONCEPT OF ROUGH SET

Rough Set is a useful mathematical tool to deal with imprecise and insufficient knowledge, reduce data sets size, find hidden patterns and generate decision rules. Rough set theory contributes immensely to the concept of reducts. Reducts is the minimal subsets of attributes with most predictive outcome. Rough sets are very effective in removing redundant features from discrete data sets.

Rough set concept is based on a pair of conventional sets called lower and upper approximations. The lower approximation is a description of objects which are known in certainty to belong to the subject of interest, while upper approximation is a description of objects which possibly belong to the subset [19].

Definition 1:

Let $S = \langle U, A, V, f \rangle$ be an information system, where U is a universe containing a finite set of N objects $\{x_1, x_2, ...x_N\}$. A is a non-empty finite set of attributes used in description of objects. V describes values of all attributes, that is, $V = \bigcup_{a \in A} V_a$ where V_a forms a set of values of the a-th attribute. $f : UxA \to V$ is the total decision function such that $f(x,a) \in V_a$ *for every* $a \in A$ *and* $x \in U$. Information system is referred to as decision table (DT) if the attributes in S is divided into two disjoint sets called condition (C) and decision attributes (D) where A = C ∪ D and C ∩ D = ϕ.

$$DT = \langle U, C \cup D, V, f \rangle \quad (1)$$

A subset of attributes $B \subseteq A$ defines an equivalent relation (called Indiscernibility relation) on U, denoted as IND(B).

$$IND(B) = \{(x, y) \in UxU \mid f(x,b) = f(y,b) \forall b \in B\} \quad (2)$$

The equivalent classes of B-indiscernibility relation are denoted $[x]_B$.

$$[x]_B = \{y \in U \mid (x, y) \in IND(B)\}$$

Definition 2: Given $B \subseteq A$ *and* $X \subseteq U$. X can be approximated using only the information contained within B by constructing the B lower and B-upper approximations of set X defined as:

$$\begin{aligned} \underline{B}X &= \{x \in X \mid [x]_B \subseteq X\} \\ \overline{B}X &= \{x \in X \mid [x]_B \cap X \neq 0\} \end{aligned} \quad (3)$$

Definition 3: Given attributes A = C ∪ D and C ∩ D = ϕ. The positive region for a given set of condition attribute C in the relation to IND (D), $POS_C(D)$ can be defined as

$$POS_C(D) = \bigcup_{x \in D^*} \underline{C}X \quad (4)$$

where D^* denotes the family of equivalence classes defined by the relation IND(D). $POS_C(D)$ contains all objects of U that can be classified correctly into
the distinct classes defined by IND(D).

Similarly, Given attributes subsets B, Q ⊆ A, the positive region contains all objects of U that can be classified to blocks of partition U/Q using attribute B. B is defined as:

$$POS_B(Q) = \bigcup_{x \in Q} \underline{B}X \quad (5)$$

Definition 4: Given attributes B, Q ⊂ A, the degree of dependency of Q on B over U is defined as

$$\gamma_B(Q) = \frac{|POS_B(Q)|}{|U|} \qquad (6)$$

The degree of dependency of an attribute dictates its significance in rough set theory.

IV. DISCRETIZATION BASED ON ENTROPY

Entropy, a supervised splitting technique used to determine how informative a particular input attribute is about the output attribute for a subset, is calculated on the basis of the class label. It is characterized by finding the split with the maximal information gain [20]. It is simply computed thus:

Let D be a set of training data set defined by a set of attributes with their corresponding labels

The Entropy for D is defined as:

$$Entropy(D) = -\sum_{i=1}^{m} P_i \log_2(P_i) \qquad (8)$$

where P_i is the probability of C_i in D, determined by dividing the number of tuples of C_i in D by |D|, the total number of tuples in D.

Given a set of samples D, if D is partitioned into two intervals D_1 and D_2 using boundary T, the entropy after partitioning is

$$E(D,T) = \frac{|D_1|}{D} Ent(D_1) + \frac{|D_2|}{D} Ent(D_2) \qquad (9)$$

where | | denotes cardinality. The boundaries T are chosen from the midpoints of the attributes values

Information gain of the split,

Gain (D,T) = Entropy(D) - E(D,T).

In selecting a spilt-point for attribute A, pick an attribute value that gives the minimum information required which is obtained when E(D,T) is minimal.. This process is performed recursively on an attribute the information requirement is less than a small threshold (0).

$$Ent(S) - E(T,S) > \delta \qquad (10)$$

V. EXPERIMENTAL SETUP AND RESULTS

The training set employed for this analysis is the “10% KDD” (kddcup_data_gz file) dataset. Since the degree of dependency is calculated for discrete features, continuous features are discretized based on entropy, discussed in section 3.1. Prior to the discretization, redundant records from the dataset were removed since rough set does not require duplicate instances to classify and identify discriminating features. For this experiment a total of 145,738 records are used, detailed shown in Table 1.

In this experiment, two approaches are adopted to detect how significant a feature is to a given class. The first approach is to compute degree of dependency for each class based on the available number of class instances in the data set. Thus, signifying how well the feature can discriminate the given class from other classes. Secondly, each class labels are mapped against others for each attribute. That is, generating a frequency table of a particular class label against others based on variations in each attribute and then a comparison made to generate the dependency ratio of predominant classes in order to detect all the relevant features distinguishing one class from another (see Appendix I for details). Graphical analysis is also employed in the analysis in order to detect the relevant features for each class.

The dependency ratio is simply computed thus

$$Dependency\ ratio = \frac{HVF}{TIN} - \frac{OTH}{TON} \qquad (11)$$

where HVF = highest number of instance variation for a class label in attribute f.
TIN = total number of instances of that class in the dataset
OTH = number of instances for other class labels based on a particular or a set of Variations.
TON = total number of instances of class labels in the data set constituting OTH

VI. RESULT DISCUSSIONS

Results are presented in terms of the class that achieved good levels of discrimination from others in the training set and the analysis of feature relevancy in the training set. Analyses are based on degree of dependency and binary discrimination for each class. That is, for each class, a dataset instance is considered in-class, if it has the same label; out-class, if it has a different label. Degree of dependency is computed for class labels based on number of instances of that class available in the dataset. Table 2 shows the highest degree of dependency of class labels depending on a particular class label in the training data set. Table 3 details the most relevant features selected for each class and their corresponding dependency ratio. Six out of the twenty three classes chooses amount of data exchange (source and destination bytes) as the most discriminating features with DOS group having half of it. This is expected of denial of service and probe category of attacks where the nature of the attack involves very short or very long connections. Feature 7 which are related for land attack is selected as the most discriminating feature for land attack while for pod and teardrop feature 8 (wrong fragment) was selected as the most discriminating features for these attack types. Also the research revealed heavy dependence on feature “Service” (i.e. feature 3) which shows that different services are exploited to perpetrate different types of attack. For instance, imap4, ftp_data and telnet are exploited to lunch imap, warezclient and buffer_overflow attack respectively. Table 4 details the most discriminating class labels for each feature. Normal, Neptune and Smurf are the most

ISBN: 978-988-17012-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Table 1: Class labels and the number of samples that appears in “10% KDD” dataset

Attack	Original Number of Samples	Number of samples after removing duplicated instances	Class
back	2,203	994	DOS
land	21	19	DOS
neptune	107,201	51,820	DOS
pod	264	206	DOS
smurf	280,790	641	DOS
teardrop	979	918	DOS
satan	1,589	908	PROBE
ipsweep	1,247	651	PROBE
nmap	231	158	PROBE
portsweep	1,040	416	PROBE
normal	97,277	87,831	NORMAL
Guess_passwd	53	53	R2L
ftp_write	8	8	R2L
imap	12	12	R2L
phf	4	4	R2L
multihop	7	7	R2L
warezmaster	20	20	R2L
warezclient	1,020	1020	R2L
spy	2	2	R2L
Buffer_overflow	30	30	U2R
loadmodule	9	9	U2R
perl	3	3	U2R
rootkit	10	10	U2R

Table 2: Attribute with the highest degree of dependency that distinctly distinguish some class labels from the training data set.

Attack	Degree of dependency	Selected features	Feature Name	Other distinct features
back	0.9708	5	source bytes	6
neptune	0.0179	3	service	39
teardrop	0.9913	8	wrong fragment	25
satan	0.0319	30	diff srv rate	27,3
portsweep	0.0264	4	flag	30,22,5
normal	0.0121	6	destination bytes	5,3,10,11,1
guess_passwd	0.0189	11	failed logins	-
imap	0.3333	26	srv error rate	-
warezmaster	0.7500	6	destination bytes	-
warezclient	0.2686	10	hot	5,1

discriminating classes for most of the features which consequently make their classification easier. Moreover, these three classes dominating the testing dataset and this account to high detection rate of machine learning algorithm on them. The research also shows how important a particular feature is to detection of an attack and normal. For some class label a feature sufficient to detect an attack type while some requires combination of two or more features. For features with few representatives in the dataset such as spy and rootkit, it is very difficult detecting a feature or features that can clearly differentiate them because of the dominance of some class labels like normal and Neptune. These difficult to classify attacks belong to two major groups, user to root and remote to local. The involvement of each feature has been analyzed for classification. Features 20 and 21 (see appendix I) make no contribution to the classification of either an attack or normal. Hence these two features (outbound command count for FTP session and hot login) have no relevance in intrusion detection. There are other features that makes little significant in the intrusion detection data set. From the dependency ratio table in Appendix I, these features include 13, 15, 17, 22 and 40 (number of compromised

Table 3: The most relevant feature for each attack type and normal conditions, su attempted, number of file creation operations, is guest login, dst host rerror rate conditions, su attempted, number of file creation operations, is guest login, dst host rerror rate respectively).

ISBN: 978-988-17012-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Table 3: The most relevant feature for each attack type and normal

Attack	Most relevant features	Feature Name	Variations	Dependency ratio	Class
Back	5	source bytes	66,64,60	0.9708	DOS
Land	7	land	2	0.9999	DOS
neptune	5	source bytes	0	0.9328	DOS
Pod	8	wrong fragment	1	0.9853	DOS
Smurf	5	source bytes	39	0.7731	DOS
teardrop	8	wrong fragment	2	0.9913	DOS
Satan	30	diff srv rate	30	0.7648	PROBE
ipsweep	36	dst host name src port rate	13,14,15,17	0.8282	PROBE
Nmap	5	source bytes	4	0.6448	PROBE
portsweep	28	srv error rate	9	0.8057	PROBE
normal	29	same srv rate	28	0.8871	NORMAL
guess_passwd	11	failed login	1	0.9622	R2L
ftp_write	23	count	1	0.7897	R2L
Imap	3	service	60	0.9980	R2L
Phf	6	destination bytes	28	0.9976	R2L
multihop	23	count	1	0.7898	R2L
warezmaster	6	destination bytes	33	0.7500	R2L
warezclient	3	service	13	0.6658	R2L
Spy	39	dst host srv serror rate	8	0.9997	R2L
buffer_overflow	3	service	6	0.6965	U2R
loadmodule	36	dst host name srcport rate	29	0.6279	U2R
Perl	14	root shell	1	0.9994	U2R
rootkit	24	srv count	1	0.7269	U2R

VII. CONCLUSION

In this paper, selection of relevance features is carried out on KDD '99 intrusion detection evaluation dataset. Empirical results revealed that some features have no relevance in intrusion detection. These features include 20 and 21 (outbound command count for FTP session and hot login) while features 13, 15, 17, 22 and 40 (number of compromised conditions, su attempted, number of file creation operations, is guest login, dst host rerror rate respectively) are of little significant in the intrusion detection.

In our future work, additional measures including sophisticated statistical tools will be employed.

Table 4: List of features for which the class is selected most relevant

Class label	Relevant features
Back	5,6
Land	7
neptune	3.4,5,23,26.29,30,31,32,34,36,37,38,39
Pod	8
Smurf	2,3,5,6,12,25,29,30,32,36,37,39
teardrop	8
Satan	27
ipsweep	36
Nmap	5
portsweep	28
normal	3,6,12,23,25,26,29,30,33,34,35,36,37,38,39
guess_passwd	11,6,3,4
ftp_write	9,23
Imap	3,39
Phf	6,10,14,5
multihop	23
warezmaster	6,1
warezclient	3,24,26
spy	39,1
buffer_overflow	3,24,14,6
loadmodule	36,24,3
perl	14,16,18,5
rootkit	24,23,3

REFERENCES

[1]. Bace, R. and Mell, P. (2001). Intrusion Detection System, NIST Special Publications SP 800. November.

[2]. Lee, W., Stolfo, S.J. and Mok, K. (1999). Data Mining in work flow environments: Experiments in intrusion detection. In Proceedings of the 1999 Conference on Knowledge Discovery and Data Mining.

[3]. Mukkamala, S., Janoski, G., Sung, A. (2002). Intrusion detection using neural networks and support vector machines. In: Proceedings of IEEE International Joint Conference on Neural Networks, pp. 1702–1707.

[4]. Byunghae, C., kyung, W.P. and Jaittyun, S. (2005) Neural Networks Techniques for Host Anomaly Intrusion Detection using Fixed Pattern Transformation in ICCSA. LNCS 3481. 254-263.

[5]. Ajith, A., Ravi J., Johnson T. and Sang, Y.H. (2005). D-SCIDS: Distributed soft computing intrusion detection system, Journal of Network and Computer Applications, Elsevier, pp. 1-19.

[6]. Susan M. B. and Rayford B.V. (2000). Intrusion detection via fuzzy data mining, Proceedings of the 12th Annual Canadian Information Technology Security Symposium, Ottawa, Canada, June 19-23, 2000, PP.109-122.

[7]. Quinlan, J.L. (1993) C4.5 Program for Machine Learning, Morgan Kaufmam Publishers, Inc.

[8]. Pavel L., Patrick D., Christia S. and Konrad R. (2005). Learning Intrusion Detection: Supervised or Unsupervised?, International Conference on image analysis and processing, (ICAP), Italie, 2005 (3617) pp. 50-57.

[9]. Zhang, L., Zhang, G., YU, L., Zhang, J. and Bai, Y. (2004) Intrusion detection using Rough Set Classification, Journal of Zhejiang University SCIENCE, 5(9):1076-1086

[10] Byung-Joo, K., and Il-Kon, K. (2005) Machine Learning Approach to Real time Intrusion Detection System in Lecture Note in Artificial Intelligence, volume 3809 Edited by S.Zhang and R.Jarvis, Springer-verlag Berline, Heidelberg, pp. 153-163.

[11].Adetunmbi, A.O., Falaki, S.O., Adewale, O.S. and Alese, B.K. (2007a) A Rough Set Approach for Detecting known and novel Network ntrusion, Second International Conference on Application of Information and Communication Technologies to Teaching, Research and Administrations (AICTTRA, 2007) eds. Kehinde, L.O., Adagunodo, E.R. and Aderounmu, G.A., OAU, Ife, pp. 190 – 200.

[12].Adetunmbi, A.O., Alese, B.K., Ogundele, O.S and Falaki, S.O. (2007b). A Data Mining Approach to Network Intrusion Detection, Journal of Computer Science & Its Applications, Vol. 14 No. 2. pp 24 -37.

[13]. Adetunmbi, A.O., Falaki, S.O., Adewale, O.S. and Alese, B.K. (2008) Intrusion Detection based on Rough Set and k-Nearest Neighbour, International Journal of Computing and ICT Research, vol. 2. pg 61-66

[14]. Sanjay, R., Gulati, V.P. and Arun, K.P. (2005) A Fast Host-Based Intrusion Detection System Using Rough Set Theory in Transactions on Rough Sets IV, LNCS 3700, 2005, pp. 144 – 161.

[15]. Axelsson, S. (1999). The Base –rate Fallacy and Its Implication for the Difficulty of Intrusion Detection, In the proceeding of the 6th ACM Conference on Computer and Communication Security. Pp. 127 -141.

[16]. Amor, N.B., Beferhat, S. and Elouedi, Z.(2004) Naïve Bayes vs Decision Trees in Intrusion Detection Systems, ACM Symposium on Applied Computing, pp. 420 – 424.

[17]. Sung, A.H. and Mukkamala, S. (2003) Identifying Important Features for Intrusion Detection using Support Vector Machines and Neural Networks, IEEE Proceedings of the 2003 Symposium on Applications and the Internet.

[18]. Kayacik, H.G., Zincir-Heywood, A.N. and Heywood, M.L. (2006). Selecting Features for Intrusion Detection: A Feature Analysis on KDD 99 Intrusion Detection Datasets.

[19]. Komorowski, J., Pokowski, L. and Skowron, A. (1998) Rough Sets: A Tutorial citeseer.ist.psu.edu/komorowski98rough.htm

[20]. Jiawei, H. and Micheline, K. (2006) Data Mining Concepts and techniques, second edition, China Machine Press, pp. 296 -303.

[21]. KDD Cup 1999 Data: http://kdd.ics.uci.edu/databases/ kddcup99/

APPENDIX I: List of Dependency ratio per attribute for each class label

Features	Normal	Neptune	Smurf	Pod	Teardrop	Land	Back	Guess_ password
1	0.0199	0.1130	0.0731	0.0729	0.0732	0.0728	0.0328	0.3303
2	-0.127	0.1401	0.9804	0.9757	0.8723	0.0897	0.0903	0.0897
3	0.6247	0.8717	0.9956	0.9763	0.6605	0.9428	0.5687	0.9967
4	0.1761	0.8080	0.0674	0.0671	0.7003	-0.0001	0.0048	0.8453
5	0.1101	0.9328	0.7731	0.9725	0.9993	0.5922	0.9964	0.7736
6	0.2126	0.8299	0.5368	0.5352	0.5357	0.5345	0.9738	0.9592
7	-0.0001	0.0000	0.0000	0.0000	0.0000	0.9999	0.0000	0.0000
8	0.0035	0.0022	0.0014	0.9767	0.9913	0.0014	0.0014	0.0014
9	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
10	0.0186	0.0211	0.0136	0.0136	0.0137	0.0136	0.9238	0.9034
11	0.0008	0.0006	0.0004	0.9622	0.0004	0.0004	0.0004	0.9622
12	0.2324	0.7582	0.4869	0.4855	0.4879	0.4848	0.2192	0.4661
13	0.0156	0.0108	0.0070	0.0070	0.0070	0.0070	0.9263	0.0070
14	0.0002	0.0005	0.0003	0.0003	0.0003	0.0003	0.0003	0.0003
15	-0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
16	-0.0063	0.0062	0.0040	0.0040	0.0040	0.0040	0.0040	0.0040
17	-0.0022	0.0028	0.0018	0.0018	0.0018	0.0018	0.0018	0.0018
18	-0.0004	0.0005	0.0003	0.0003	0.0003	0.0003	0.0003	0.0003
19	-0.0049	0.0048	0.0031	0.0031	0.0031	0.0031	0.0031	0.0031
20	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
21	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
22	0.0012	0.0073	0.0047	0.0047	0.0047	0.0047	0.0047	0.0112
23	0.2901	0.4786	0.2069	0.2554	0.1436	0.6845	0.4150	0.2804
24	0.1485	0.0795	0.2061	0.2203	0.2671	0.6897	0.2098	0.3909
25	0.7343	0.0007	0.3016	0.3007	0.1169	0.6521	0.2831	0.2437
26	0.7165	0.8071	0.2999	0.2990	0.3005	0.7031	0.2794	0.2420
27	0.1409	0.1318	0.1128	0.1125	0.0283	0.0596	0.1353	0.8017
28	0.1320	0.1271	0.1155	0.1152	0.1157	0.1151	0.1843	0.7983
29	0.8871	0.4239	0.3824	0.3812	0.1431	0.2755	0.3743	0.3809
30	0.8864	0.7226	0.3828	0.3816	0.0749	0.2759	0.3747	0.3813
31	0.1359	0.3549	0.2306	0.2299	0.2310	0.7359	0.0740	0.2297
32	0.0906	0.6247	0.3244	0.2038	0.1988	0.8274	0.0476	0.3937
33	0.2746	0.2357	0.0221	0.0771	0.1451	0.3772	0.0573	0.2678
34	0.6437	0.5434	0.2787	0.0509	0.2058	0.3958	0.5575	0.5538
35	0.6439	0.3857	0.2841	0.1629	0.1974	0.4012	0.5629	0.5592
36	0.2081	0.6750	0.6881	0.4035	0.1553	0.8034	0.2021	0.2377
37	0.2525	0.5401	0.3505	0.1206	0.3512	0.2067	0.3514	0.3302
38	0.6990	0.8087	0.2640	0.1439	0.2007	0.5444	0.1834	0.5432
39	0.6736	0.8704	0.3308	0.3298	0.3314	0.2627	0.1796	0.2260
40	0.1224	0.1422	0.0689	0.0861	0.1873	0.0515	0.2787	0.5456
41	0.1073	0.1443	0.1403	0.1399	0.1406	0.1397	0.2783	0.7499

ISBN: 978-988-17012-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)