

A Granular Intrusion Detection System using Rough Cognitive Networks

Peer-reviewed author version

NAPOLES RUIZ, Gonzalo; Grau, Isel; Falcon, Rafael; Bello, Rafael & VANHOOF, Koen (2016) A Granular Intrusion Detection System using Rough Cognitive Networks. In: Abielmona, Rami; Falcon, Rafael; Zincir-Heywood, Nur; Abbass, Hussein A. (Ed.). Recent Advances in Computational Intelligence in Defence and Security, p. 169-191.

DOI: 10.1007/978-3-319-26450-9_7

Handle: <http://hdl.handle.net/1942/21458>

See discussions, stats, and author profiles for this publication at:
<https://www.researchgate.net/publication/283505893>

A Granular Intrusion Detection System Using Rough Cognitive Networks

Chapter · January 2016

DOI: 10.1007/978-3-319-26450-9_7

CITATION

1

READS

51

5 authors, including:



Rafael Falcon

Larus Technologies Corpora...

69 PUBLICATIONS 246 CITATIONS

SEE PROFILE



Koen Vanhoof

Hasselt University

243 PUBLICATIONS 1,998
CITATIONS

SEE PROFILE

A Granular Intrusion Detection System Using Rough Cognitive Networks

Gonzalo Nápoles, Isel Grau, Rafael Falcon, Rafael Bello
and Koen Vanhoof

1 **Abstract** Security in computer networks is an active research field since traditional
2 approaches (e.g., access control, encryption, firewalls, etc.) are unable to completely
3 protect networks from attacks and malwares. That is why Intrusion Detection Sys-
4 tems (IDS) have become an essential component of security infrastructure to detect
5 these threats before they inflict widespread damage. Concisely, network intrusion
6 detection is essentially a pattern recognition problem in which network traffic pat-
7 terns are classified as either normal or abnormal. Several Computational Intelligence
8 (CI) methods have been proposed to solve this challenging problem, including fuzzy
9 sets, swarm intelligence, artificial neural networks and evolutionary computation.
10 Despite the relative success of such methods, the complexity of the classification
11 task associated with intrusion detection demands more effective models. On the other
12 hand, there are scenarios where identifying abnormal patterns could be a challenge

G. Nápoles · I. Grau · R. Bello

Computer Science Department, Central University of Las Villas, Carretera
Camajuaní Km 5.5, 54830 Santa Clara, Cuba
e-mail: gnapoles@uclv.edu.cu

I. Grau

e-mail: igráu@uclv.edu.cu

R. Bello

e-mail: rbellop@uclv.edu.cu

R. Falcon (✉)

Research & Engineering Division, Larus Technologies Corporation,
170 Laurier Ave. West - Suite 310, Ottawa, ON K1P 5V5, Canada
e-mail: rafael.falcon@larus.com; rfalcon@uottawa.ca

R. Falcon

Electrical Engineering & Computer Science, University of Ottawa,
800 King Edward Ave., Ottawa, ON K1N 6N5, Canada

G. Nápoles · K. Vanhoof

Hasselt Universiteit Campus Diepenbeek, Agoralaan Gebouw D,
BE3590 Diepenbeek, Belgium
e-mail: gonzalo.napoles@student.uhasselt.be

K. Vanhoof

e-mail: koen.vanhoof@uhasselt.be

© Springer International Publishing Switzerland 2016

R. Abielmona et al. (eds.), *Recent Advances in Computational Intelligence
in Defense and Security*, Studies in Computational Intelligence 621,
DOI 10.1007/978-3-319-26450-9_7

13 as the collected data is still permeated with uncertainty. In this chapter, we tackle the
14 network intrusion detection problem from a classification angle by using a recently
15 proposed granular model named Rough Cognitive Networks (RCN). An RCN is a
16 fuzzy cognitive map that leans upon rough set theory to define its topological con-
17 structs. An optimization-based learning mechanism for RCNs is also introduced.
18 The empirical evidence indicates that the RCN is a suitable approach for detecting
19 abnormal traffic patterns in computer networks.

20 **Keywords** Intrusion detection system · Computational intelligence · Granular
21 computing · Rough set theory · Fuzzy cognitive maps · Rough cognitive networks ·
22 Harmony search

23 1 Introduction

24 The 21st century has brought forth a digital age in which we are all immersed.
25 Up-and-coming information communication and processing paradigms such as the
26 Internet of Things (IoT) [4], Cloud Computing [47], Software-Defined Networks
27 [32] and Wearable Computing [25] are increasingly gaining momentum and rapidly
28 permeating every facet of mankind. These new architectural frameworks bring a
29 unique set of challenges with them, among which cybersecurity is one of para-
30 mount importance. The computer systems that constitute the backbone of critical
31 infrastructure behind a plethora of industrial and societal processes often become
32 prey to sophisticated malicious attacks that originate at any node in the entangled
33 World Wide Web. As a result, governments and businesses are adapting their leg-
34 islative bodies to account for the prevention, detection and mitigation of the risks
35 and threats associated with these potentially devastating attacks [39].

36 Intrusion Detection Systems (IDS) [43] have become an essential component of
37 security infrastructure to detect these threats before they inflict widespread dam-
38 age, since traditional approaches (e.g., access control, encryption, firewalls, etc.) are
39 unable to completely protect networks from attacks and malwares. The purpose of an
40 IDS is to analyze the network traffic, either the incoming one or existing logs of past
41 traffic activities, and identify anomalous behaviours that could reasonably be taken
42 as cues of the presence of an intruder in the system. Concisely described, network
43 intrusion detection is essentially a pattern recognition problem in which network
44 traffic patterns are classified as either normal or abnormal.

45 Although traditional statistical techniques have enjoyed success in analyzing traf-
46 fic flows as part of an IDS operation, the network security research community is
47 increasingly leaning on Computational Intelligence (CI) solutions due to their ability
48 to adapt to complex environments, handle noise and uncertainty and remain compu-
49 tationally tractable and robust.

50 More recently, the advent of Granular Computing (GrC) [6, 26, 52] as an innov-
51 ative information representation and processing framework has largely influenced
52 the way CI systems are being conceived nowadays. This is due to the fact that

53 GrC provides reasoning constructs at higher levels of abstraction that better capture
54 human understanding of the real world. From classification [55] to clustering [51],
55 time-series prediction [72] and decision making [50], granular models are becoming
56 prominent tools for the analysis of large volumes of data as they operate upon
57 information granules (i.e., constructs of order higher than plain numeric or symbolic
58 atoms) and can better represent and manifest the dynamics of human-centric world
59 modeling.

60 In this chapter, we tackle network intrusion detection via a GrC model and demon-
61 strate its advantages over several traditional classification schemes. Our study makes
62 the following contributions: (1) we model network intrusion detection as a classifi-
63 cation problem and apply a recently introduced granular model, named “Rough Cog-
64 nitive Network”(RCN), to the analysis of archived traffic data in computer networks
65 for intrusion detection purposes; (2) we put forth a learning mechanism for RCNs
66 that is based on self-adaptive Harmony Search [44]; (3) we empirically evaluate the
67 RCN performance in conjunction with that of seven well-established classifiers in
68 the literature. The experimental evidence confirms that RCNs are a plausible model
69 to discriminate between normal and abnormal traffic patterns in network data as it
70 attains high detection rates (i.e., successfully identified abnormalities) and low false
71 negative rates (misidentified anomalies).

72 The rest of this chapter is structured as follows. Section 2 briefly surveys relevant
73 works in intrusion detection systems, with special emphasis on CI-based solutions.
74 Section 3 elaborates on the two precursor formalisms leading up to RCNs: rough set
75 theory (RST) and fuzzy cognitive maps (FCMs). Then, the RCN topology learning
76 and classification inference process are dissected in Sect. 4 while Sect. 5 describes
77 the proposed optimization-based RCN parameter learning method. The experimen-
78 tal analysis is unveiled in Sect. 6 before conclusions and future work directions are
79 outlined in Sect. 7.

80 2 Related Work

81 In this section, we briefly review several published works that are relevant to our
82 study. They provide the necessary background to understand the contents of this
83 chapter.

84 2.1 Intrusion Detection Systems

85 The literature in the IDS arena is quite vast. This field appears often interwoven
86 with other similar terms such as “network anomaly detection” or “network intrusion
87 detection” and the common underlying problem has been addressed through a myriad
88 of techniques. In a recent and comprehensive survey [8] covering publications in
89 this field from 2000 to 2012, 28 % of the papers surveyed approached IDS from

90 a supervised learning angle (i.e., classification), as we do in this chapter. However,
 91 unsupervised learning (via clustering) was the preferred choice of 21 % of the papers
 92 given that labeled data could be scarce and/or difficult to access in certain cases
 93 where privacy concerns impede the sharing of such information.

94 The statistical methods and systems applied to intrusion detection [45, 61, 66,
 95 79] first construct a general statistic model of the observed traffic data, either via
 96 parametric techniques (which assume the knowledge of the type of probability
 97 distribution is available and then try to learn their parameters) or by means of non-
 98 parametric techniques, which do not lay any assumption on the type of the data
 99 distribution. Once this model has been fitted to the data, any point (traffic pat-
 100 tern) with low probability of having been generated by the underlying data model
 101 is labeled as an outlier and hence flagged as suspicious.

102 The use of computational intelligence methods in the IDS realm has been well
 103 documented in the 2010 survey compiled by Wu and Banzhaf [73]. Artificial neural
 104 networks (ANNs) [11, 40, 67, 78, 81], fuzzy sets [16, 21, 29, 68], evolutionary com-
 105 putation [5, 18, 24, 31, 38, 57–59], artificial immune systems (AIS) [70, 75], fuzzy
 106 cognitive maps [62–64, 74, 83], rough sets [2, 13, 14] and swarm intelligence (SI)
 107 [19, 20, 29] techniques, all representative methods of the wider CI/Soft Computing
 108 (SC) family, and their hybrids [15, 22, 63, 64, 74] have all been wielded against
 109 complex network traffic datasets to identify attack vectors or suspicious activities
 110 either in a supervised or unsupervised fashion.

111 *2.2 Rough Set Theory in Network Security*

112 Rough sets and fuzzy cognitive maps have been independently applied to network
 113 intrusion detection [8, 73], although the number of reported works thus far is not sig-
 114 nificant compared to the volume of documented applications of other CI techniques.

115 Chen et al. [13] employ rough set theory in the preprocessing stage of their pro-
 116 posed network intrusion detection scheme in order to remove irrelevant attributes
 117 prior to the operation of the Support Vector Machine (SVM)-based classifier. A
 118 similar use (attribute dimensionality reduction) is evoked by Li and Zhao with their
 119 Fuzzy SVM [41] and by Zhang et al. in the context of their Artificial Immune System
 120 (AIS)-based technique [82], where the number of attributes that describe an antibody
 121 is shortened using the lower and upper approximations of each rough concept. Shri-
 122 vastava and Jain [60] also boost the network traffic classification power of their SVM
 123 via rough-set-based feature selection by dropping 35 irrelevant attributes out of 41
 124 initially gathered to describe the traffic flows in their system. An analogous ratio-
 125 nale is pursued by Sivaranjanadevi et al. in their work [65] and by Poongothai and
 126 Duraiswamy in [53].

127 Fuzzy and rough sets are integrated into a partitive clustering engine in [14] to
128 address network intrusion detection from an unsupervised perspective; the proposed
129 clustering method yielded superior results compared to other classical unsupervised
130 techniques.

131 Finally, rough sets are used in [2] to induce classification rules via the LEM2
132 algorithm so as to create a potent classifier capable of detecting network intrusions
133 with high detection rate and low false alarm rate. The classification results of LEM2
134 are found to be more interpretable and can be obtained in a shorter time than those
135 of the K-nearest neighbor classifier, which are more accurate yet more resource-
136 demanding.

137 *2.3 Fuzzy Cognitive Maps in Network Security*

138 Xin et al. [74] derive fuzzy features from the network data and pass them on to a
139 fuzzy cognitive map (FCM) in order to model more complex attack vectors.

140 Siraj et al. [63] used FCM and fuzzy rule bases to model causal knowledge
141 among different intrusion variables in an interpretable fashion. Suspicious events
142 are mapped to nodes in FCM, which function as neurons that trigger alerts with dif-
143 ferent weights depicting on the causal relations between them. So, an alert value for a
144 particular machine or a user is calculated as a function of all the activated suspicious
145 events at a given time. This value reflects the safety level of that machine or user at
146 that time.

147 Siraj et al. [64] chose FCMs and fuzzy rule bases as the vehicles for causal knowl-
148 edge acquisition within the decision engine of an intelligent IDS deployed at the
149 Mississippi State University. The system fuses information from a variety of intru-
150 sion detection sensors. In particular, the FCMs are used at two levels: (i) to model
151 individual suspicious events such as ‘high login failure’ or ‘SYN flood’ and (ii) to
152 ascertain the overall impact of various suspicious events (input concepts) for each
153 host computer and system user (output concepts).

154 Afterwards, Siraj and Vaughn [62] also leaned upon FCMs to cluster network
155 intrusion alerts based on discovered similarities among the raw features extracted
156 from sensor data. The FCM is thus acting as a fusion machine where intrusion evi-
157 dence for a particular network resource that originates at different clusters is amal-
158 gamated.

159 Zhong et al. [83] consider a distributed attack scenario and resort to an FCM to
160 describe the entities that are part of it as well as their relationships.

161 The study authored by Jazzar and Bin Jantan [27] focuses on IDS designed around
162 the Self Organizing Map (SOM) neural network given its ability to process large vol-
163 umes of data with low computational overhead. Having realized that these systems
164 still exhibit a high false alarm rate, they coupled the SOM with an FCM in order
165 to refine the clustering performed by the former approached. The FCM’s role is to
166 calculate the relevance of odd concepts (neurons) to a network attack. By doing so,

167 irrelevant concepts can be left out and other concepts may come to the forefront of
168 the intrusion analysis.

169 Krichene and Boudriga [37] devised a methodology to automatically determine
170 responses to security incidents. The underlying formalism that allows attack identifi-
171 cation, complexity reduction and response elicitation is termed an *incident response*
172 *probabilistic cognitive map*. These maps differ from traditional FCMs in that they are
173 capable of modeling different relationships between symptoms, actions and unautho-
174 rized results as pertaining to a network attack. A function that enables the identifi-
175 cation of those concepts that are tied to a set of events is also part of the proposed
176 scheme. The authors illustrate their proposal on a real-world denial of service (DoS)
177 attack against a web server.

178 Zaghoud and Al-Kahtani [80] bring forth a multi-layered architecture for intru-
179 sion detection and response. They employ an FCM to gauge the impact of a con-
180 firmed intrusion event belonging to a known class upon the compromised system.
181 The FCM nodes represent components of the computer network system or security
182 concepts whereas the edges symbolize the influence exercised by one component
183 upon another; these influences must be carefully taken into consideration now that a
184 network intrusion has been confirmed.

185 2.4 Discussion

186 Our proposed granular classifier, the Rough Cognitive Network, borrows from both
187 aforementioned techniques: RST and FCM; however, their synergy is dictated by a
188 topological arrangement of the FCM nodes into symbolic and higher-order informa-
189 tion granules, the latter of which correspond to the three RST-based regions (posi-
190 tive, boundary, negative) of the decision concepts (classes) induced by a similarity
191 relationship over the set of input attributes in the data set under consideration. To the
192 best of our knowledge, this hybridization scheme is completely different from previ-
193 ous efforts to combine both methodologies, and so is certainly the RCN application
194 to the IDS domain.

195 3 The Forerunners of Rough Cognitive Networks

196 As mentioned before, in this paper we design an IDS which uses an RCN for detecting
197 potentially atypical (and likely dangerous) patterns. One could briefly define an RCN
198 as a Sigmoid Fuzzy Cognitive Map where concepts represent granules of informa-
199 tion. In this section, we summarize the mathematical underpinnings behind Rough
200 Set Theory and Fuzzy Cognitive Maps, which are the two core building blocks of
201 the granular model proposed in this chapter.

202 3.1 Rough Set Theory

203 Rough Set Theory (RST) is a robust and mature theory for handling uncertainty
 204 in the form of inconsistency in the data [1, 49]. The RST framework employs two
 205 exact set approximations to describe a generic or real-world concept. Let us assume
 206 a decision system $S = (U, A \cup d)$, where U is a non-empty finite set of objects called
 207 the universe, A is a non-empty finite set of attributes, while $d \notin A$ denotes the deci-
 208 sion attribute. Any subset $X \subseteq U$ can be approximated by two crisp sets: the lower
 209 and upper approximations. These sets are defined as $B_*X = x \in U : [x]_B \subseteq X$ and
 210 $B^*X = x \in U : [x]_B \cap X \neq \emptyset$ where the equivalence class $[x]_B$ comprises the set of
 211 inseparable objects associated to the target instance x that are described using $B \subseteq A$.

212 Based on the lower and upper approximations, we can compute the positive,
 213 negative and boundary regions of any concept X . The positive region $POS(X) =$
 214 B_*X includes those objects that are certainly contained in X ; the negative region
 215 $NEG(X) = U - B^*X$ involves those objects that are certainly not contained in X ,
 216 whereas the boundary region $BND(X) = B^*X - B_*X$ represents the objects whose
 217 membership to the set X is uncertain, i.e., they might be members of X . These
 218 regions are in fact information granules and provide a valuable knowledge when
 219 facing decision-making or pattern classification problems.

220 Based on the positive, negative and boundary regions, Yao [76] defined two types
 221 of rules: *deterministic* decision rules for the positive region and *undeterministic*
 222 decision rules for the boundary region. More recently Yao [77] introduced the three-way
 223 decisions model. Rules constructed from the three regions are associated with differ-
 224 ent actions [23]. A positive rule suggests a decision of *acceptance*, a negative rule
 225 makes a decision of *rejection* and a boundary rule implies a decision of *abstaining*.
 226 The three-way decisions play an important role in decision-making problems [42].

227 In the classical RST formulation, the indiscernibility relation is defined as an
 228 equivalence relation; hence, two objects will be inseparable if they are identical with
 229 respect to a set of attributes $B \subseteq A$. The equivalence relation R induces a partition of
 230 the universe U on the basis of the attributes in B . However, this definition is extremely
 231 strict. For example, a decision system with millions of objects will be categorized
 232 as inconsistent if two objects are equivalent but they have different decision classes
 233 (i.e., two experts might have different perceptions about the same observation). But
 234 are two objects really significant in a universe comprised of millions of objects?

235 To counter the above stringent definition, the equivalence requirement on R is
 236 relaxed. In fact, if we adopt a “weaker” inseparability relation then we could tackle
 237 problems having numerical (or mixed) attributes. Two inseparable objects, according
 238 to some similarity relationship R , will be tossed together in the same set of not identi-
 239 cal (but reasonably similar) instances. Equation 1 shows the indiscernibility relation
 240 adopted in this paper, where $0 \leq \varphi(x, y) \leq 1$ is a similarity function. This binary rela-
 241 tion determines whether two objects x and y are inseparable or not (i.e., as long as
 their similarity degree $\varphi(x, y)$ is greater than or equal to a user-specified threshold ξ).

242 Despite the clear advantages of using this approach to cope with problems having
 243 numerical features, selecting the correct value for the similarity threshold ξ could be
 244 a challenge.

$$245 \quad R : yRx \Leftrightarrow \varphi(x, y) \geq \xi. \quad (1)$$

246 If the threshold $\xi = 1$ then the similarity relation R will be reflexive, transitive
 247 and symmetric, leading to Pawlak's model for discrete (nominal) domains. If $\xi < 1$
 248 then the similarity relation will be reflexive and symmetric but not transitive.

249 Another aspect to be considered when designing a similarity relation is the ade-
 250 quate selection of the similarity function. Equation 2 shows a variant which combines
 251 both numerical and categorical attributes. It provides a more general formulation for
 252 addressing decision-making problems having different features.

$$253 \quad \varphi(x, y) = \frac{1}{|A|} \sum_{i=1}^{|A|} \omega_i \delta(x(i), y(i)). \quad (2)$$

254 In the above equation, A is the set of features describing the problem, $0 \leq \omega_i \leq 1$
 255 represents the relative importance of the i th attribute, $x(i)$ and $y(i)$ denote the val-
 256 ues of the i th attribute associated with the objects x and y respectively, and δ is the
 257 attribute-wise similarity function. The greater $0 \leq \varphi(x, y) \leq 1$, the more similar the
 258 objects x and y . Equations 3 and 4 display the attribute-wise similarity functions
 259 adopted in this research study. The function δ_1 is used when we want to compare
 260 two values of a discrete attribute, whereas δ_2 is used for comparing two values of a
 261 numerical attribute (L_i and H_i denote the lowest and highest value of the i th attribute,
 262 respectively).

$$263 \quad \delta_1(x(i), y(i)) = \begin{cases} 1, & x(i) = y(i) \\ 0, & x(i) \neq y(i) \end{cases}. \quad (3)$$

$$264 \quad \delta_2(x(i), y(i)) = 1 - \frac{|x(i) - y(i)|}{H_i - L_i}. \quad (4)$$

266 Equations 5 and 6 respectively formalize how to compute the lower and upper
 267 approximations of a concept X , where $R(x)$ denotes the similarity class of the object
 268 x . These exact sets are the basis for granulating the available information about the
 269 concept using RST, and they become the core of Granular Fuzzy Cognitive Maps
 270 [48].

$$271 \quad B_*X = \{x \in U : R(x) \subseteq X\}. \quad (5)$$

$$272 \quad B^*X = \bigcup_{x \in X} R(x). \quad (6)$$

274 As a result, an object can simultaneously belong to multiple similarity classes,
 275 so the covering induced by the similarity relation R over the universe U is not nec-
 276 essarily a partition [7]. Therefore, similarity relations do not induce a partition of

277 the universe, but rather generate similarity classes. It suggests that an object could
 278 simultaneously belong to different similarity classes, and consequently the instance
 279 x could activate several granular regions. In such cases, the decision-making stage
 280 becomes really difficult for the expert, since it has to consider non-trivial decision
 281 patterns.

282 3.2 Fuzzy Cognitive Maps

283 Fuzzy Cognitive Maps (FCM) are recurrent neural networks for modeling and simu-
 284 lation [34] consisting of concepts and their causal relations. Concepts are equivalent
 285 to neurons denoting objects, variables, or entities related to the system under inves-
 286 tigation whereas the weights associated with the connections among neurons denote
 287 the strength of the *causality* among such nodes. It should be highlighted that causal
 288 relations are quantified in the range $[-1; 1]$. This value is the result of the numerical
 289 evaluation of a fuzzy linguistic variable, which is usually assigned by experts during
 290 the modeling phase [36]. The activation value of the neurons is also fuzzy in nature
 291 and regularly takes values in the range $[0; 1]$ although the interval $[-1; 1]$ is used
 292 too. The magnitude of the activation is also meaningful for the model: the higher the
 293 activation value of a map concept, the stronger its influence over the system under
 294 consideration.

295 Equation 7 mathematically formalizes the rule for updating the activation value of
 296 concepts in an FCM, assuming A^0 is the initial configuration. This rule is iteratively
 297 repeated until a fixed point attractor or a maximum number of iterations T is reached.
 298 At each step t a new state vector is produced, and after a large enough number of
 299 iterations, the map will arrive at one of the following states: (i) fixed equilibrium
 300 point, (ii) limited cycle or (iii) chaotic behavior [35]. If the FCM reaches a fixed-
 301 point attractor, then we can conclude that the map has converged. In such cases,
 302 the final output corresponds to the desired state (i.e., the system response for the
 303 activation vector).

$$304 \quad A_i^{t+1} = f\left(\sum_{j=1}^M w_{ji}A_j^t + w_{ii}A_i^t\right), i \neq j. \quad (7)$$

305 In the above equation $f(\cdot)$ represents a monotonically non-decreasing nonlinear
 306 function which is used for transforming the activation value of each concept (the
 307 weighted combination of the activation levels). The most used functions are: the
 308 bivalent function, the trivalent function, and sigmoid variants [10]. In this paper we
 309 will focus on sigmoid functions since it has been shown that they exhibit superior
 310 prediction capabilities [10].

311 4 Rough Cognitive Networks

312 Rough Cognitive Networks (RCNs) [48] are an extension of three-way decision rules
 313 introduced by Yao [76]. In a nutshell, we can define an RCN as a sigmoid FCM where
 314 concepts denote information granules, namely, the RST-derived positive, boundary
 315 and negative regions of the original problem as well as the set of decision classes in
 316 the problem at hand.

317 The RCN methodology not just allows solving mixed-attribute problems, but also
 318 provides accurate inferences since it uses a recurrent inferential process to converge
 319 to a stable attractor, which comprises the most fitting decision class. It should be
 320 pointed out that the complexity of this model does not depend on the number of
 321 attributes in the decision system, but on the number of decision classes. In this
 322 section, we explain how to learn an RCN from data. Furthermore, we introduce a
 323 supervised learning algorithm for computing the required RCN parameters, which
 324 enhances the value of our proposal.

325 4.1 Information Granulation and Network Design

326 As mentioned before, a central aspect when designing an RCN is the process related
 327 to the construction of positive, negative and boundary regions. Let us assume a pat-
 328 tern classification problem and a partition $X = X_1, \dots, X_k, \dots, X_N$ of the universe
 329 U according to the decision attribute, where each subset X_k denotes a decision
 330 class and comprises all instances labeled as d_k . These information granules will be
 331 expressed as map concepts. More precisely, input concepts denote positive, negative
 332 and boundary regions associated with each subset X_k ; they are subsequently used for
 333 activating the network.

334 In the RCN model, the output neurons do not influence other neurons since they
 335 are target concepts. Once the FCM inference process is done (this point will be
 336 clarified next), the activation degree of each output concept (decision class) will be
 337 gauged. After the map concepts are defined, we establish causal connections among
 338 such neurons, where the direction and intensity of the causal weights are computed
 339 according to the set of rules below:

- 340 • R_1 : IF C_i is P_k AND C_j is d_k THEN $w_{ij} = 1.0$.
- 341
- 342 • R_2 : IF C_i is P_k AND C_j is $d_{(v \neq k)}$ THEN $w_{ij} = -1.0$.
- 343
- 344 • R_3 : IF C_i is P_k AND C_j is $P_{(v \neq k)}$ THEN $w_{ij} = -1.0$.
- 345
- 346 • R_4 : IF C_i is N_k AND C_j is d_k THEN $w_{ij} = -1.0$.

347 In the above rules, C_i and C_j denote two map concepts, P_k and N_k are the positive
 348 and negative region for the k th decision respectively, whereas $-1 \leq w_{ij} \leq 1$ is the

causal weight between the cause C_i and the effect C_j . More precisely, rules R_1 and R_2 define the relation between positive regions and decision neurons. If the positive region P_k is activated (rule 1), then the decision d_k must be stimulated as well, since we confidently know that objects belonging to the positive region P_k will be categorically members of the concept X_k . Accordingly, decisions $d_{(v \neq k)}$ must be inhibited (rule 2) because an object cannot simultaneously belong to different positive regions.

The third rule follows an analogous reasoning: if a positive region P_k is activated then positive regions unrelated to the decision d_k (i.e., $P_{(v \neq k)}$) will be inhibited. If the negative region N_k is activated (rule 4), then the map will inhibit the decision, but we cannot conclude anything about other decisions. Moreover, we incorporated an additional rule for handling the intrinsic knowledge concerning the RST boundary regions:

- R_5 : IF C_i is B_k AND C_j is d_v AND $(BND(X_k) \cap BND(X_v) \neq \emptyset)$ THEN $w_{ij} = 0.5$.

Observe that not all boundary regions are included in the RCN's topology. This is dictated by the learning procedure on the training data: if a boundary region is empty ($BND(X_k) = \emptyset$) then the neuron B_k will be removed from the modeling in order to simplify the network topology. On the other hand, we need to establish causal links between each boundary neuron and decision classes involving some degree of uncertainty; otherwise the causal connection will be removed from the map as well.

The above topology construction scheme implies that an RCN for a problem with $|D|$ decision classes will have at most $3|D|$ input neurons (assuming all boundary regions are in), $|D|$ decision (output) neurons and $3|D|(1 + |D|)$ causal relations. Additionally, for each neuron we add a self-reinforcement connection with causality $w_{ii} = 1$ which partially preserves the initial excitation.

4.2 Inference Using Rough Cognitive Networks

The final phase concerns the network exploitation, where the activation value of input and decision concepts play a pivotal role. In this scheme, to classify a test instance O_i , first the excitation vector A_i will be calculated using the similarity class $R(O_i)$ and its relation to each RST-based region. For instance, let us assume that $|POS(X_1)| = 20$, $|R(O_i)| = 10$, whereas the number of objects that belong to the positive region is given by the expression: $|R(O_i) \cap POS(X_k)| = 7$. This implies that the activation degree of the neuron P_1 is $7/20 = 0.35$. It denotes the conditional probability of accepting d_1 given the similarity class $R(O_i)$, that is $Pr(d_1 | R(O_i))$. Analogously, we can compute the activation degree of other input concepts related to each decision class. Rules $R_6 - R_8$ formalize this procedure as follows:

- R_6 : IF C_i is P_k THEN $A_i^0 = \frac{|R(O_i) \cap POS(X_k)|}{|POS(X_k)|}$.
- R_7 : IF C_i is N_k THEN $A_i^0 = \frac{|R(O_i) \cap NEG(X_k)|}{|NEG(X_k)|}$.
- R_8 : IF C_i is B_k THEN $A_i^0 = \frac{|R(O_i) \cap BND(X_k)|}{|BND(X_k)|}$.

387 Once the activation vector A^0 has been computed, we trigger the FCM inference
 388 rule until a fixed point attractor, or a maximal number of iterations T is reached. This
 389 process will stress a pattern using the similarity class of the instance O_i to do that,
 390 which is desirable in problems with insufficient positive evidence where selecting
 391 the proper class could be difficult. Afterward one can use the output vector for making
 392 a decision (e.g., we can sort the alternatives according to the preference degrees
 393 calculated by the map inference process). When dealing with pattern classification
 394 problems, the final output will be the concept having the highest activation, or alter-
 395 natively it could be a random class if the input similarity class only activates negative
 396 and/or boundary regions.

397 5 Learning Methodology for Rough Cognitive Networks

398 As mentioned before, the basis for computing the set of positive, negative and bound-
 399 ary regions is the proper estimation of the similarity threshold ξ in Eq. 1. If this value
 400 is too small then positive regions will be small as well, leading to poor excitation of
 401 neurons. This step is quite important when selecting the most adequate decision: the
 402 higher the activation of the positive region, the more desirable the decision (although
 403 the model will compute the final decision taking into account all the evidence). If this
 404 threshold ξ is excessively large then boundary regions will be large, thus increasing
 405 the uncertainty.

406 In this section, we present a learning algorithm for tuning the model parameters,
 407 which is based on the Harmony Search (HS) metaheuristic [44]. The method needs
 408 to adjust two kinds of parameters: the weight ω_i of each attribute and the similarity
 409 threshold ξ . This approach leads to a numerical optimization problem with $|A| + 1$
 410 variables and will be solved using an adaptive variant of the HS procedure.

411 The HS metaheuristic is a simple-trajectory search method, which only evaluates
 412 one potential solution at a time, instead of evaluating a set of potential solutions (as
 413 it occurs with population-based metaheuristics). This HS design choice is relevant
 414 for our learning methodology since evaluating a solution means computing the set
 415 of lower and upper approximations, which could be computationally expensive as
 416 the number of objects in the training data set increase.

417 During the optimization phase, the algorithm randomly creates a harmony mem-
 418 ory with size HMS and iteratively improves a new harmony from the HM. If the
 419 improved harmony is better than the worst harmony in the HM, then the new solu-
 420 tion replaces the worst harmony. Despite its algorithmic simplicity, HS suffers from
 421 a serious problem common to other metaheuristics: its search capabilities are quite
 422 sensitive to the specified parameter vector.

423 For this reason in this paper we adopt an improved variant, called Self-adaptive
 424 Harmony Search (SHS), which is capable of adjusting its own parameters [71].
 425 The SHS method not only alleviates the parametric sensitivity issue, but also sig-
 426 nificantly enhances the accuracy of the solutions. Algorithm 1 shows the pseudo-
 427 code of this metaheuristic, where N is the maximal number of iterations, HMCR

428 (Harmony Memory Consideration Rate) is a parameter that controls the balance
 429 between exploitations and exploration, while $R_1 = U - x$ and $R_2 = x - L$, assuming
 430 that L and U respectively denote the lowest and the highest values for each problem
 431 variable in the harmony memory.

432 On the other hand, PAR is the pitch adjustment rate and determines whether fur-
 433 ther adjustment is required to a harmony drawn from the harmony memory. In this
 434 variant, the PAR factor is linearly decreased over time. Experiments reported by the
 435 authors [71] suggested that moderate size of the harmony memory (e.g., 50) and
 436 large values of HMCR (e.g., 0.9) are adequate choices for these parameters. Based
 437 on these considerations, we used these values during the experiments and simula-
 438 tions performed in the next section. The $rand()$ function draws a random number
 439 uniformly distributed in the unit interval.

440 **Algorithm 1. Self-adaptive Harmony Search**

```

441 Initialize the memory
442 FOR  $i = 1$  TO  $N$  DO
443   IF  $rand() < HMCR$  THEN
444     Select a random pitch  $x$  from the memory
445     IF  $rand() < PAR$  THEN
446        $x = x + rand(R_1, R_2)$ 
447     END
448   ELSE
449      $x = x + rand(a, b)$ 
450   END
451   Select the worst harmony  $y$  from the memory
452   IF ( $y$  is worse than  $x$ ) THEN
453     Replace the worst harmony  $y$  with the new pitch  $x$ 
454   END
455 END
456 Select the best solution  $S$  from the memory
457 RETURN  $S$ 
458
459

```

460 The other component of the optimization problem to be specified is the objective
 461 function. Equation 8 shows the function $G(\cdot)$ used in this study, where the parameters
 462 denote the set of weights W , the similarity threshold ξ and the set of instances ϕ to
 463 be used for training the model, respectively. On the other hand, $\mathfrak{N}_{R(W, \xi)}(x)$ is the
 464 output vector computed by the RCN which is obtained from the similarity threshold
 465 defined by the function $R(W, \xi)$, whereas the function $Y(x)$ is the known class vector
 466 associated with the instance x and D is the set of decision classes in the problem. It
 467 should be also mentioned that $\|\cdot\|_L$ refers to a norm (e.g., the L_1 -norm, L_2 -norm or
 468 L_∞ -norm) that is used to calculate the error.

$$469 \quad \text{minimize } G(W, \xi, \phi) = \sum_{x \in \phi} \frac{\|\mathfrak{N}_{R(W, \xi)}(x) - Y(x)\|_L}{|\phi||D|}. \quad (8)$$

470 If $G(W, \xi, \phi) = 0$ then the RCN, using the similarity relation R , is capable of
 471 recognizing all patterns stored in the training set; otherwise the value $1 - G(W, \xi, \phi)$
 472 stands for the model accuracy. The proposed parameter tuning method not only esti-
 473 mates the introduced parameters, but also allows determining the relevance of each
 474 attribute, which contributes to elicit further knowledge about the problem.

475 6 Detecting Intrusion in Computer Networks

476 In this section we study the performance of the proposed granular cognitive network
 477 for detecting abnormal traffic behavior in computer networks. As mentioned before,
 478 this problem can be envisioned as a challenging pattern classification task having two
 479 decision classes: either ‘normal’ or ‘abnormal’. In order to perform our simulations,
 480 we used an improved variant of the NSL-KDD dataset [17] which is a widely used
 481 benchmark when testing IDS [19, 22, 23]. In the following section, we summarize
 482 the most important features of both training and testing NSL-KDD datasets.

483 6.1 Description of the NSL-KDD Dataset

484 Perhaps the most popular dataset for evaluating the performance of anomaly detec-
 485 tion models is KDD’99 [30]. The KDD training dataset consists of 4,900,000 net-
 486 work connection vectors, each of which contains 41 features. Such features could
 487 be gathered in three groups: (i) basic features, (ii) traffic features and (iii) content
 488 features.

489 The first group comprises attributes extracted from a TCP/IP connection, whereas
 490 the second one includes time-based features computed in a window interval (e.g.,
 491 connections in the past 2 s having the same destination host or the same service
 492 as the current connection). It should be stated that there are several slow-probing
 493 attacks that scan the ports using a much larger time interval than 2 s and accordingly
 494 these attacks will not produce any intrusion patterns. Finally, the third group contains
 495 features related to attacks having a single connection, which do not have intrusion
 496 frequent sequential patterns. In such cases, attacks are embedded in the data por-
 497 tions of packets, hence forcing the Intrusion Detection System to catch suspicious
 498 behavior in the data portion (e.g., number of failed login attempts) instead of in the
 499 connections.

500 On the other hand, in the training set each record is labeled as either “normal”
 501 or “abnormal” with exactly one specific attack type (i.e., Probing Attack, Denial of
 502 Service Attack, User to Root Attack and Remote to Local Attack).

503 It is essential to mention that the KDD’99 dataset was built based on the data
 504 captured in DARPA’98 which has been criticized by McHugh [46]. It suggests that
 505 some of the existing problems in the dataset DARPA’98 remain in KDD’99. More
 506 recently, Tavallae and collaborators [69] conducted a statistical analysis where two

507 important issues were detected. The first important deficiency in the KDD'99 dataset
 508 is the huge number of redundant records (78 and 75 % of records are duplicated in
 509 the train and test set, respectively). Consequently, this will cause learning algorithms
 510 to be biased towards the more frequent records. As a second issue they noticed that
 511 this dataset has poor difficulty level: about 98 % of the records in the train set and
 512 86 % of the records in the test set were correctly classified with 21 learned machines
 513 (7 learners, each trained 3 times with different training sets).

514 To solve the aforementioned issues, Tavallae et al. [69] removed all the redun-
 515 dant records in both train and test sets. Moreover, they randomly sampled correctly
 516 classified records in such a way that the number of selected instances from each
 517 difficulty level group is inversely proportional to the percentage of records in the
 518 original dataset. This refinement process gave rise to two improved datasets called
 519 KDDTrain+ and KDDTest+ which include 125,973 and 22,544 records, respec-
 520 tively. As well, they created another test set called KDDTest-21 by removing the
 521 records that were correctly classified by all 21 learners. This dataset contains 11,850
 522 records, which are more difficult to classify. Because of its increasing popularity and
 523 sound verification procedure, we adopted Tavallae et al's data sets for our experi-
 524 mentation.

525 6.2 Numerical Simulations

526 Next we study the behavior of RCN across the selected dataset. Figure 1 displays
 527 the network topology that allows solving the prediction problem (i.e., where each

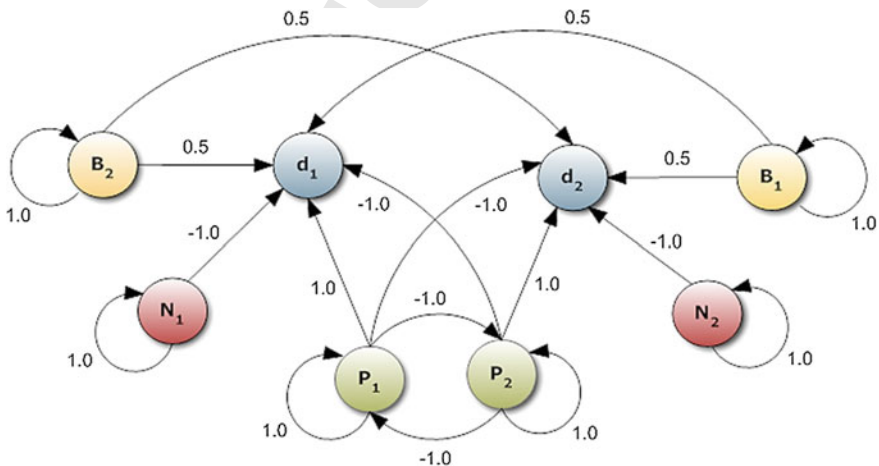


Fig. 1 The proposed Rough Cognitive Network for intrusion detection. The d_1 concept corresponds to the normal traffic class and the d_2 concept represents the abnormal traffic class. The P_i , B_i and N_i nodes denote the positive, boundary and negative regions for these two classes, $i \in \{1, 2\}$

instance is classified as either “normal” or “abnormal”). More exactly, $d_1 =$ “normal”, $d_2 =$ “abnormal”, P_i denotes the positive region associated to the i th class, N_i is the negative region related to the i th class while B_i is the i th boundary region. Note that boundary concepts are allowed regardless of the inconsistency of the features in the target problem because only two decision classes are possible. More explicitly, if the problem has inconsistent instances, then both classes will be equally affected; otherwise, the activation value of the (empty) boundary regions will remain inactive during the inference process.

6.2.1 Comparison with Traditional Classifiers Over KDDTest+

The first experiment consists of studying the prediction ability of our model regarding the following set of traditional classifiers: J48 decision tree [54], NBTree [33], Random Forest [9], Random Tree [3], Multilayer Perceptron [56], Naive Bayes [28], and Support Vector Machine [12]. For experimental purposes, we adopted the first 20% of the records in KDDTrain+ for training all models. Figure 2a summarizes the accuracy achieved for each learner, whereas Fig. 2b displays some representative samples of the solution space associated with the similarity threshold to be explored by the learning algorithm. In other words, Fig. 2b illustrates the performance of our granular network for different similarity thresholds.

From the above experiment we can conclude that RCN results are competitive regarding J48 decision tree, Random Forest (RF), NBTree (NBT) and Random Tree (RT). However, our model outperforms other approaches such as Multilayer Perceptron (MLP), Naive Bayes (NB) and Support Vector Machine (SVM).

Next we study other statistics such as those extracted from the confusion matrices. True Negatives (TN) as well as True Positives (TP) correspond to correctly classified instances, that is, events that are rightly labeled as normal and attacks, respectively. Alternatively, False Positives (FP) refer to normal events being labeled as attacks while False Negatives (FN) are attack events incorrectly predicted as normal events. Table 1 shows such statistics for all classifiers used for comparison across the selected KDDTest+ dataset.

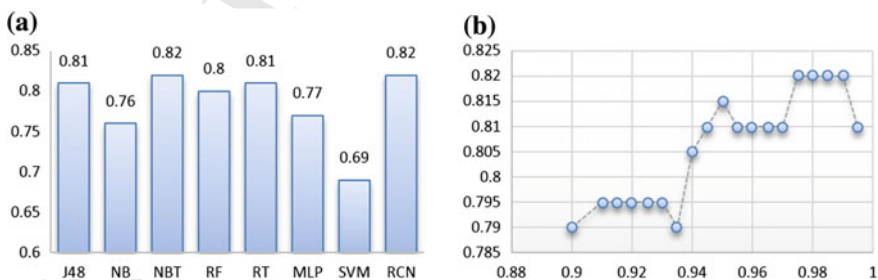


Fig. 2 Experiments using datasets KDDTrain+ and KDDTest+. **a** Accuracy of selected classifiers and **b** RCN accuracy as a function of the threshold values in Eq. (1)

Table 1 Confusion matrix associated with each classifier for the KDDTest+ dataset

	TN	FP	FN	TP	Detection rate	False alarm rate
J48	9436	275	3996	8837	0.68	0.02
NB	9010	701	4582	8251	0.64	0.07
NBT	8869	842	3257	9576	0.74	0.08
RF	9452	259	4523	8310	0.64	0.02
RT	8898	813	3011	9822	0.76	0.08
MLP	8971	740	4796	8037	0.62	0.07
SVM	8984	727	4893	7940	0.61	0.07
RCN	8891	820	3150	9683	0.75	0.08

557 The reader may notice that RCN ranks as the second-best algorithm regarding
 558 the number of FN patterns. In our study we are especially interested in this value
 559 since it denotes the number of abnormal patterns that the IDS was unable to detect,
 560 although most authors prefer systems with high detection rate (i.e., $TP/(TP + FN)$)
 561 and low false alarm rate which is defined as $FP/(TN + FP)$. Nevertheless, in computer
 562 networks where high security is required, reducing the false negative rate is
 563 indispensable since only those patterns having normal features will be confidently
 564 allowed.

565 6.2.2 Comparison with Traditional Classifiers Over KDDTest-21

566 The second experiment is concerned with investigating the performance of our
 567 RCN model with respect to traditional classifiers, but now using the test set called
 568 KDDTest-21. Figure 3a portrays the classification accuracy achieved for each model
 569 while Fig. 3b displays the performance of the proposed granular network for different
 570 similarity thresholds.

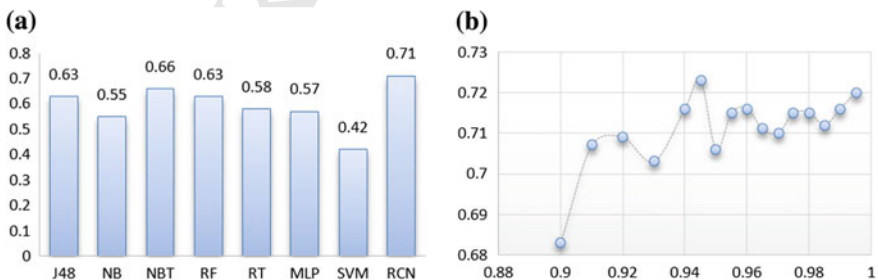


Fig. 3 Experiments using datasets KDDTrain+ and KDDTest-21. **a** Accuracy of selected classifiers and **b** RCN accuracy as a function of the threshold values in Eq. (1)

Table 2 Confusion matrix associated with each classifier for the KDDTest-21 dataset

	TN	FP	FN	TP	Detection rate	False alarm rate
J48	1879	273	3996	5702	0.58	0.12
NB	1460	692	4549	5149	0.53	0.32
NBT	1354	798	3257	6441	0.66	0.37
RF	1895	257	4523	5175	0.53	0.11
RT	1388	764	3008	6690	0.68	0.35
MLP	1426	726	4796	4902	0.50	0.33
SVM	1440	712	4893	4805	0.49	0.33
RCN	1572	580	2824	6874	0.70	0.26

571 It should be specified that the KDDTest-21 dataset is more complex since it
572 involves patterns that cannot be correctly classified by all learners. Despite this fact,
573 our model was able to compute the best accuracy (71 %), notably outperforming the
574 remaining approaches. However, in a previous experiment the model only achieved
575 an accuracy of 66 % due to the uncertainty present in the features during the inference
576 stage (i.e., the overall evidence suggests accepting both decisions). To overcome
577 this situation, we used the similarity classes pertaining to the K -nearest neighbors
578 ($K = 3$) of the test instance O_i . In short, we adopted the similarity classes of its neighbors
579 instead of only using the set $R(O_i)$ related to the target pattern for activating each
580 input neuron in the network.

581 Table 2 shows the confusion matrix achieved by each classifier across the
582 KDDTest-21 test set. In this case, our model computed the highest detection rate
583 ($TP/(TP + FN) = 0.7$) and lowest false negative rate ($FN/(TP + FN) = 0.29$) which
584 is the desired behavior. It means that the RCN will detect abnormal traffic with high
585 accuracy, thus reducing the risk of classifying abnormal patterns as normal. In a nutshell,
586 such statistics confirm the reliability of our granular classifier (RCN) for intrusion
587 detection in complex computer networks. For instance, the reader may observe
588 that if the false alarm rate is high, then the system will classify normal patterns as
589 abnormal, but this behavior is preferable in order to avoid potential attacks.

590 6.2.3 Discussion

591 Although the above experiments show that RCNs are a suitable approach for addressing
592 intrusion detection problems, there are cases where the inference suggests accepting
593 a wrong decision class. This behavior could be a direct result of the strategy
594 adopted for activating the input concepts, so other ways for estimating the activation
595 vector could be explored. For example, in Bayesian inference one usually translates
596 $Pr(C|[x])$ into $Pr([x|C]Pr(C))/Pr([x])$ by the Bayes theorem, which allows a practical
597 estimation of initial conditions required to trigger the FCM inference process.

598 Another aspect to be considered is related to the network weights, since rules
599 R_1-R_5 formalize the direction (negative or positive) of each causal connection rather
600 than its intensity. This means that the granular neural network discussed in this
601 chapter calculates the decision class based on the initial state A^0 and the sign of
602 causal relations, without exploiting the causal intensity. To achieve further perfor-
603 mance gains, we are currently focused on computing this indicator via a supervised
604 learning approach.

605 7 Conclusions

606 An important aspect in computer networks is how to detect intrusion since traditional
607 approaches such as access control lists or firewalls are incapable of entirely protect-
608 ing networks. In order to deal with such problem, several intrusion detection systems
609 have been proposed; however, increasing the overall performance (e.g., the detec-
610 tion accuracy) is still an open problem for researchers. More explicitly, an essential
611 component of intrusion detection systems is the inference algorithm used to classify
612 network traffic patterns as either normal or abnormal. This problem could be thought
613 of as a challenging binary classification task since modern intrusion techniques are
614 sophisticated, so it is difficult to design models being able to distinguish between
615 normal and abnormal patterns. As an example, frequently hackers attempt simulat-
616 ing trusted users in computer networks in order to gain access to remote resources.
617 Such behavior will produce inconsistency in the collected traffic data; that is, objects
618 that are very similar yet have been labeled as pertaining to different decision classes.

619 In this chapter we introduced a novel IDS based on Rough Cognitive Networks,
620 a recently proposed granular neural network for pattern classification. Without loss
621 of generality, we can define RCN as a Sigmoid Fuzzy Cognitive Map where input
622 neurons represent information granules whereas output concepts denote decision
623 classes. It should be remarked that the granulation of information is achieved by
624 using Rough Sets, since it allows handling uncertainty arising from inconsistency.
625 Furthermore, with the goal of increasing the reliability of the RCN-based inference
626 process, we discussed a supervised learning methodology for automatically comput-
627 ing accurate similarity relations by estimating the proper parameter vector.

628 In order to measure the performance of our model, we adopted an improved ver-
629 sion of the NSL-KDD dataset. From numerical simulations it is possible to conclude
630 that our granular neural network is a suitable approach for detecting abnormal traffic
631 patterns in computer networks. More precisely, we observed that RCNs are com-
632 petitive regarding traditional classifiers such as J48 decision tree and Random For-
633 est, across the simpler dataset (KDDTrain+). However, for the dataset KDDTest-21
634 the model significantly outperformed the other learners by computing the highest
635 detection rate ($DR = 0.7$) and lowest false negative rate ($FNR = 0.29$). This con-
636 firms the reliability of the learning methodology put forth in this chapter to boost
637 the model's performance. Future work along this front will concentrate on validat-
638 ing our approach on real computer networks.

References

- 640 1. Abraham, A., Falcon, R., Bello, R.: *Rough Set Theory: A True Landmark in Data Analysis*.
641 Springer, Heidelberg (2009)
- 642 2. Adetunmbi, A.O., Falaki, S.O., Adewale, O.S., Alese, B.K.: Network intrusion detection based
643 on rough set and k-nearest neighbour. *I. J. Comput. ICT Res.* **2**(1), 60–66 (2008)
- 644 3. Aldous, D.: The continuum random tree. *I. Ann. Prob.* 1–28 (1991)
- 645 4. Atzori, L., Iera, A., Morabito, G.: The internet of things: a survey. *Comput. Netw.* **54**(15),
646 2787–2805 (2010)
- 647 5. Balajinath, B., Raghavan, S.: Intrusion detection through learning behavior model. *Comput.*
648 *Commun.* **24**(12), 1202–1212 (2001)
- 649 6. Bello, R., Falcon, R., Pedrycz, W., Kacprzyk, J.: *Granular Computing: At The Junction of*
650 *Rough Sets and Fuzzy Sets*. Springer, Heidelberg (2008)
- 651 7. Bello, R., Verdegay, J.L.: Rough sets in the soft computing environment. *Inf. Sci.* **212**, 1–14
652 (2012)
- 653 8. Bhuyan, M.H., Bhattacharyya, D., Kalita, J.K.: Network anomaly detection: methods, systems
654 and tools. *IEEE Commun. Surv. Tutorials* **16**(1), 303–336 (2014)
- 655 9. Breiman, L.: Random forests. *Mach. Learn.* **45**(1), 5–32 (2001)
- 656 10. Bueno, S., Salmeron, J.L.: Benchmarking main activation functions in fuzzy cognitive maps.
657 *Expert Syst. Appl.* **36**(3), 5221–5229 (2009)
- 658 11. Cannady, J.: Artificial neural networks for misuse detection. In: *National Information Systems*
659 *Security Conference*, pp. 368–81 (1998)
- 660 12. Chang, C.C., Lin, C.J.: Libsvm: a library for support vector machines. *ACM Trans. Intell. Syst.*
661 *Technol. (TIST)* **2**(3), 27 (2011)
- 662 13. Chen, R.C., Cheng, K.F., Chen, Y.H., Hsieh, C.F.: Using rough set and support vector machine
663 for network intrusion detection system. In: *First Asian Conference on Intelligent Information*
664 *and Database Systems, 2009. ACIIDS 2009*, pp. 465–470. IEEE (2009)
- 665 14. Chimphee, W., Abdullah, A.H., Noor Md Sap, M., Srinoy, S., Chimphee, S.: Anomaly-based
666 intrusion detection using fuzzy rough clustering. In: *International Conference on Hybrid Infor-*
667 *mation Technology, 2006. ICHIT'06*, vol. 1, pp. 329–334. IEEE (2006)
- 668 15. Costa, K.A., Pereira, L.A., Nakamura, R.Y., Pereira, C.R., Papa, J.P., Falcão, A.X.: A nature-
669 inspired approach to speed up optimum-path forest clustering and its application to intrusion
670 detection in computer networks. *Inf. Sci.* **294**, 95–108 (2015)
- 671 16. Dickerson, J.E., Dickerson, J.A.: Fuzzy network profiling for intrusion detection. In: *19th*
672 *International Conference of the North American Fuzzy Information Processing Society, 2000.*
673 *NAFIPS*, pp. 301–306. IEEE (2000)
- 674 17. Elkan, C.: Results of the KDD'99 classifier learning. *ACM SIGKDD Explor. Newsl.* **1**(2),
675 63–64 (2000)
- 676 18. Faraoun, K., Boukelif, A.: Genetic programming approach for multi-category pattern classifi-
677 cation applied to network intrusions detection. *Int. J. Comput. Intell. Appl.* **6**(01), 77–99 (2006)
- 678 19. Feng, W., Zhang, Q., Hu, G., Huang, J.X.: Mining network data for intrusion detection through
679 combining svms with ant colony networks. *Future Gener. Comput. Syst.* **37**, 127–140 (2014)
- 680 20. Gao, H.H., Yang, H.H., Wang, X.Y.: Ant colony optimization based network intrusion feature
681 selection and detection. In: *Proceedings of 2005 International Conference on Machine Learn-*
682 *ing and Cybernetics, 2005*, vol. 6, pp. 3871–3875. IEEE (2005)
- 683 21. Geramiraz, F., Memaripour, A.S., Abbaspour, M.: Adaptive anomaly-based intrusion detection
684 system using fuzzy controller. *Int. J. Netw. Secur.* **14**(6), 352–361 (2012)
- 685 22. Govindarajan, M.: Hybrid intrusion detection using ensemble of classification methods. *Int. J.*
686 *Comput. Netw. Inf. Secur.* **2**, 45–53 (2014)
- 687 23. Guo, C., Zhou, Y., Ping, Y., Zhang, Z., Liu, G., Yang, Y.: A distance sum-based hybrid method
688 for intrusion detection. *Appl. Intell.* **40**(1), 178–188 (2014)
- 689 24. Hofmann, A., Schmitz, C., Sick, B.: Rule extraction from neural networks for intrusion detec-
690 tion in computer networks. In: *IEEE International Conference on Systems, Man and Cybernet-*
691 *ics, 2003*, vol. 2, pp. 1259–1265. IEEE (2003)



- 692 25. Hong, J., Baker, M.: Wearable computing. *IEEE Pervasive Comput.* **13**(2), 7–9 (2014)
- 693 26. Jankowski, A., Skowron, A.: Toward perception based computing: A rough-granular perspec-
694 tive. In: Zhong, N., Liu, J., Yao, Y., Wu, J., Lu, S., Li, K. (eds.) *Web Intelligence Meets Brain*
695 *Informatics. Lecture Notes in Computer Science*, vol. 4845, pp. 122–142. Springer, Heidelberg
696 (2007)
- 697 27. Jazzar, M., Bin Jantan, A.: Using fuzzy cognitive maps to reduce false alerts in SOM-based
698 intrusion detection sensors. In: *Second Asia International Conference on Modeling Simulation*,
699 2008. AICMS 08, pp. 1054–1060 (2008)
- 700 28. John, G.H., Langley, P.: Estimating continuous distributions in Bayesian classifiers. In: *Pro-*
701 *ceedings of the Eleventh conference on Uncertainty in artificial intelligence*, pp. 338–345.
702 Morgan Kaufmann Publishers Inc. (1995)
- 703 29. Karami, A., Guerrero-Zapata, M.: A fuzzy anomaly detection system based on hybrid pso-
704 kmeans algorithm in content-centric networks. *Neurocomputing* **149**, 1253–1269 (2015)
- 705 30. KDD Cup 1999: KDD'99 dataset (2007). [http://kdd.ics.uci.edu/databases/kddcup99/](http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html)
706 [kddcup99.html](http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html)
- 707 31. Khan, M.S.A.: Rule based network intrusion detection using genetic algorithm. *Int. J. Comput.*
708 *Appl.* **18**(8), 26–29 (2011)
- 709 32. Kirkpatrick, K.: Software-defined networking. *Commun. ACM* **56**(9), 16–19 (2013)
- 710 33. Kohavi, R.: Scaling up the accuracy of Naive-Bayes classifiers: a decision-tree hybrid. In:
711 KDD, pp. 202–207 (1996)
- 712 34. Kosko, B.: Fuzzy cognitive maps. *Int. J. Man Mach. Stud.* **24**(1), 65–75 (1986)
- 713 35. Kosko, B.: Hidden patterns in combined and adaptive knowledge networks. *Int. J. Approximate*
714 *Reasoning* **2**(4), 377–393 (1988)
- 715 36. Kosko, B.: *Fuzzy Engineering* (1996)
- 716 37. Krichene, J., Boudriga, N.: Incident response probabilistic cognitive maps. In: *International*
717 *Symposium on Parallel and Distributed Processing with Applications*, 2008. ISPA '08, pp.
718 689–694 (2008)
- 719 38. Kuang, F., Xu, W., Zhang, S.: A novel hybrid KPCA and SVM with GA model for intrusion
720 detection. *Appl. Soft Comput.* **18**, 178–184 (2014)
- 721 39. Kuehn, A.: *Extending Cybersecurity, Securing Private Internet Infrastructure: the US Einstein*
722 *Program and its Implications for Internet Governance*. Springer (2014)
- 723 40. Labib, K., Vemuri, V.R.: NSOM: A tool to detect denial of service attacks using self-organizing
724 maps. Department of Applied Science University of California, Davis, California, USA, Tech-
725 nical Report (2002)
- 726 41. Li, L., Zhao, K.: A new intrusion detection system based on rough set theory and fuzzy support
727 vector machine. In: *2011 3rd International Workshop on Intelligent Systems and Applications*
728 *(ISA)*, pp. 1–5 (2011)
- 729 42. Liang, D., Pedrycz, W., Liu, D., Hu, P.: Three-way decisions based on decision-theoretic rough
730 sets under linguistic assessment with the aid of group decision making. *Appl. Soft Comput.*
731 **29**, 256–269 (2015)
- 732 43. Liu, G.G.: *Intrusion detection systems*. In: *Applied Mechanics and Materials*, vol. 596, pp.
733 852–855. Trans Tech Publications (2014)
- 734 44. Loganathan, G.: A new heuristic optimization algorithm: harmony search. *Simulation* **76**(2),
735 60–68 (2001)
- 736 45. Manikopoulos, C., Papavassiliou, S.: Network intrusion and fault detection: a statistical anomaly
737 approach. *IEE Commun. Mag.* **40**(10), 76–82 (2002)
- 738 46. McHugh, J.: Testing intrusion detection systems: a critique of the 1998 and 1999 DARPA
739 intrusion detection system evaluations as performed by Lincoln laboratory. *ACM Trans. Inf.*
740 *Syst. Secur.* **3**(4), 262–294 (2000)
- 741 47. Mell, P., Grance, T.: *The NIST definition of cloud computing* (2011)
- 742 48. Nápoles, G., Grau, I., Vanhoof, K., Bello, R.: Hybrid model based on rough sets theory and
743 fuzzy cognitive maps for decision-making. In: *Kryszkiewicz, M., Cornelis, C., Ciucci, D.,*
744 *Medina-Moreno, J., Motoda, H., Ras, Z. (eds.) RSEISP 2014* (2014)
- 745 49. Pawlak, Z.: Rough sets. *Int. J. Comput. Inf. Sci.* **11**(5), 341–356 (1982)

- 746 50. Pedrycz, W., Al-Hmouz, R., Morfeq, A., Balamash, A.S.: Building granular fuzzy decision
747 support systems. *Knowl.-Based Syst.* **58**, 3–10 (2014)
- 748 51. Pedrycz, W., Al-Hmouz, R., Morfeq, A., Balamash, A.S.: Distributed proximity-based granular
749 clustering: towards a development of global structural relationships in data. *Soft Comput.* 1–17
750 (2014)
- 751 52. Pedrycz, W., Skowron, A., Kreinovich, V.: *Handbook of Granular Computing*. Wiley (2008)
- 752 53. Poongothai, T., Duraiswamy, K.: Effective cross layer intrusion detection in mobile ad hoc
753 networks using rough set theory and support vector machines. *Asian J. Inf. Technol.* **12**(8),
754 242–249 (2013)
- 755 54. Quinlan, J.R.: *C4.5: Programs for Machine Learning* (2014)
- 756 55. Roh, S.B., Pedrycz, W., Ahn, T.C.: A design of granular fuzzy classifier. *Expert Syst. Appl.*
757 **41**(15), 6786–6795 (2014)
- 758 56. Ruck, D.W., Rogers, S.K., Kabrisky, M., Oxley, M.E., Suter, B.W.: The multilayer perceptron
759 as an approximation to a Bayes optimal discriminant function. *IEEE Trans. Neural Netw.* **1**(4),
760 296–298 (1990)
- 761 57. Shafi, K., Abbass, H.A.: Biologically-inspired complex adaptive systems approaches to net-
762 work intrusion detection. *Inf. Secur. Tech. Rep.* **12**(4), 209–217 (2007)
- 763 58. Shafi, K., Abbass, H.A.: An adaptive genetic-based signature learning system for intrusion
764 detection. *Expert Syst. Appl.* **36**(10), 12036–12043 (2009)
- 765 59. Shafi, K., Kovacs, T., Abbass, H.A., Zhu, W.: Intrusion detection with evolutionary learning
766 classifier systems. *Nat. Comput.* **8**(1), 3–27 (2009)
- 767 60. Shrivastava, S.K., Jain, P.: Effective anomaly based intrusion detection using rough set theory
768 and support vector machine. *Int. J. Comput. Appl.* **18**(3), 35–41 (2011)
- 769 61. Simmross-Wattenberg, F., Asensio-Pérez, J.I., Casaseca-de-la H.P., Martin-Fernandez, M.,
770 Dimitriadis, I.A., Alberola-Lopez, C.: Anomaly detection in network traffic based on statistical
771 inference and alpha-stable modeling. *IEEE Trans. Dependable Secure Comput.* **8**(4), 494–509
772 (2011)
- 773 62. Siraj, A., Vaughn, R.: Multi-level alert clustering for intrusion detection sensor data. In: *Annual*
774 *Meeting of the North American Fuzzy Information Processing Society, 2005. NAFIPS 2005*,
775 pp. 748–753 (2005)
- 776 63. Siraj, A., Bridges, S.M., Vaughn, R.B.: Fuzzy cognitive maps for decision support in an intel-
777 ligent intrusion detection system. In: *Joint 9th IFSA World Congress and 20th NAFIPS Inter-*
778 *national Conference, 2001, vol. 4*, pp. 2165–2170. IEEE (2001)
- 779 64. Siraj, A., Vaughn, R.B., Bridges, S.M.: Intrusion sensor data fusion in an intelligent intrusion
780 detection system architecture. In: *Proceedings of the 37th Annual Hawaii International Con-*
781 *ference on System Sciences, 2004*, pp. 1–10. IEEE (2004)
- 782 65. Sivaranjanadevi, P., Geetanjali, M., Balaganesh, S., Poongothai, T.: An effective intrusion sys-
783 tem for mobile ad hoc networks using rough set theory and support vector machine. *IJCA Proc.*
784 *E Governance Cloud Comput. Serv.* **2**, 1–7 (2012)
- 785 66. Song, X., Wu, M., Jermaine, C., Ranka, S.: Conditional anomaly detection. *IEEE Trans. Knowl.*
786 *Data Eng.* **19**(5), 631–645 (2007)
- 787 67. Sun, J., Yang, H., Tian, J., Wu, F.: Intrusion detection method based on wavelet neural network.
788 In: *Second International Workshop on Knowledge Discovery and Data Mining, 2009. WKDD*
789 *2009*, pp. 851–854. IEEE (2009)
- 790 68. Tajbakhsh, A., Rahmati, M., Mirzaei, A.: Intrusion detection using fuzzy association rules.
791 *Appl. Soft Comput.* **9**(2), 462–469 (2009)
- 792 69. Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A.: A detailed analysis of the KDD CUP 99
793 data set. In: *Proceedings of the Second IEEE Symposium on Computational Intelligence for*
794 *Security and Defence Applications 2009* (2009)
- 795 70. Visconti, A., Tahayori, H.: Artificial immune system based on interval type-2 fuzzy set para-
796 digm. *Appl. Soft Comput.* **11**(6), 4055–4063 (2011)
- 797 71. Wang, C.M., Huang, Y.F.: Self-adaptive harmony search algorithm for optimization. *Expert*
798 *Syst. Appl.* **37**(4), 2826–2837 (2010)

- 799 72. Wang, W., Pedrycz, W., Liu, X.: Time series long-term forecasting model based on information
800 granules and fuzzy clustering. *Eng. Appl. Artif. Intell.* **41**, 17–24 (2015)
- 801 73. Wu, S.X., Banzhaf, W.: The use of computational intelligence in intrusion detection systems:
802 a review. *Appl. Soft Comput.* **10**(1), 1–35 (2010)
- 803 74. Xin, J., Dickerson, J., Dickerson, J.A.: Fuzzy feature extraction and visualization for intrusion
804 detection. In: *The 12th IEEE International Conference on Fuzzy Systems, 2003. FUZZ'03*,
805 vol. 2, pp. 1249–1254. IEEE (2003)
- 806 75. Yang, H., Li, T., Hu, X., Wang, F., Zou, Y.: A survey of artificial immune system based intru-
807 sion detection. *Sci. World J.* **2014** (2014)
- 808 76. Yao, Y.: Three-way decision: An interpretation of rules in rough set theory. *Lecture Notes in*
809 *Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture*
810 *Notes in Bioinformatics)* 5589 LNAI, 642–649 (2009)
- 811 77. Yao, Y.: Three-way decisions with probabilistic rough sets. *Inf. Sci.* **180**(3), 341–353 (2010)
- 812 78. Yong, H., Feng, Z.X.: Expert system based intrusion detection system. In: *2010 International*
813 *Conference on Information Management, Innovation Management and Industrial Engineering*
814 *(ICIII)*, vol. 4, pp. 404–407. IEEE (2010)
- 815 79. Yu, M.: A nonparametric adaptive cusum method and its application in network anomaly detec-
816 tion. *Int. J. Advancements Comput. Technol.* **4**(1), 280–288 (2012)
- 817 80. Zaghdoud, M., Al-Kahtani, M.S.: Contextual fuzzy cognitive map for intrusion response sys-
818 tem. *Int. J. Comput. Inf. Technol.* **2**(3), 471–478 (2013)
- 819 81. Zhang, C., Jiang, J., Kamel, M.: Comparison of BPL and RBF network in intrusion detection
820 system. In: *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing*, pp. 466–470.
821 Springer (2003)
- 822 82. Zhang, L., Bai, Z., Luo, S., Cui, G., Li, X.: A dynamic artificial immune-based intrusion detec-
823 tion method using rough and fuzzy set. In: *2013 International Conference on Information and*
824 *Network Security (ICINS 2013)*, pp. 1–7 (2013)
- 825 83. Zhong, C., Yang, F., Zhang, L., Li, Z.: An efficient distributed coordinated intrusion detection
826 algorithm. In: *2005 International Conference on Machine Learning and Cybernetics*, pp. 2679–
827 2685 (2006)