http://www.icact.org/journal.asp

ICACT-TACT JOURNAL

Transactions on Advanced Communications Technology

Volume 4 Issue 4, Jul. 2015, ISSN: 2288-0003

Editor-in-Chief
Prof. Thomas Byeongnam YOON, PhD.

Global IT
Research Institute

Journal Editorial Board

Editor Guide

■ Introduction for Editor or Reviewer

All the editor group members are to be assigned as a evaluator(editor or reviewer) to submitted journal papers at the discretion of the Editor-in-Chief. It will be informed by eMail with a Member Login ID and Password.

Once logined the Website via the Member Login menu in left as a evaluator, you can find out the paper assigned to you. You can evaluate it there. All the results of the evaluation are supposed to be shown in the Author Homepage in the real time manner. You can also enter the Author Homepage assigned to you by the Paper ID and the author's eMail address shown in your Evaluation Webpage. In the Author Homepage, you can communicate each other efficiently under the peer review policy. Please don't miss it!

All the editor group members are supposed to be candidates of a part of the editorial board, depending on their contribution which comes from history of ICACT TACT as an active evaluator. Because the main contribution comes from sincere paper reviewing role.

■ Role of the Editor

The editor's primary responsibilities are to conduct the peer review process, and check the final camera-ready manuscripts for any technical, grammatical or typographical errors.

As a member of the editorial board of the publication, the editor is responsible for ensuring that the publication maintains the highest quality while adhering to the publication policies and procedures of the ICACT TACT(Transactions on the Advanced Communications Technology).

For each paper that the editor-in-chief gets assigned, the Secretariat of ICACT Journal will send the editor an eMail requesting the review process of the paper.

The editor is responsible to make a decision on an "accept", "reject", or "revision" to the Editor-in-Chief via the Evaluation Webpage that can be shown in the Author Homepage also.

■ Deadlines for Regular Review

Editor-in-Chief will assign a evalaution group(a Editor and 2 reviewers) in a week upon receiving a completed Journal paper submission. Evaluators are given 2 weeks to review the paper. Editors are given a week to submit a recommendation to the Editor-in-Chief via the evaluation Webpage, once all or enough of the reviews have come in. In revision case, authors have a maximum of a month to submit their revised manuscripts. The deadlines for the regular review process are as follows:

Evalution Procedure	Deadline
Selection of Evaluation Group	1 week
Review processing	2 weeks
Editor's recommendation	1 week
Final Decision Noticing	1 week

■ Making Decisions on Manuscript

Editor will make a decision on the disposition of the manuscript, based on remarks of the reviewers. The editor's recommendation must be well justified and explained in detail. In cases where the revision is requested, these should be clearly indicated and explained. The editor must then promptly convey this decision to the author. The author may contact the editor if instructions regarding amendments to the manuscript are unclear. All these actions could be done via the evaluation system in this Website. The guidelines of decisions for publication are as follows:

Decision	Description
Accept	An accept decision means that an editor is accepting the paper with no further modifications. The paper will not be seen again by the editor or by the reviewers.
Reject	The manuscript is not suitable for the ICACT TACT publication.
Revision	The paper is conditionally accepted with some requirements. A revision means that the paper should go back to the original reviewers for a second round of reviews. We strongly discourage editors from making a decision based on their own review of the manuscript if a revision had been previously required.

■ Role of the Reviewer

Reviewer Webpage:

Once logined the Member Login menu in left, you can find out papers assigned to you. You can also login the Author Homepage assigned to you with the paper ID and author's eMail address. In there you can communicate each other via a Communication Channel Box.

Quick Review Required:

You are given 2 weeks for the first round of review and 1 week for the second round of review. You must agree that time is so important for the rapidly changing IT technologies and applications trend. Please respect the deadline. Authors undoubtedly appreciate your quick review.

Anonymity:

Do not identify yourself or your organization within the review text.

Review:

Reviewer will perform the paper review based on the main criteria provided below. Please provide detailed public comments for each criterion, also available to the author.

- How this manuscript advances this field of research and/or contributes something new to the literature?
- Relevance of this manuscript to the readers of TACT?
- Is the manuscript technically sound?
- Is the paper clearly written and well organized?
- Are all figures and tables appropriately provided and are their resolution good quality?
- Does the introduction state the objectives of the manuscript encouraging the reader to read on?
- Are the references relevant and complete?

Supply missing references:

Please supply any information that you think will be useful to the author in revision for enhancing quality of the paperor for convincing him/her of the mistakes.

Review Comments:

If you find any already known results related to the manuscript, please give references to earlier papers which contain these or similar results. If the reasoning is incorrect or ambiguous, please indicate specifically where and why. If you would like to suggest that the paper be rewritten, give specific suggestions regarding which parts of the paper should be deleted, added or modified, and please indicate how.

Journal Procedure

Dear Author,

- **You can see all your paper information & progress.**

- **Step 1. Journal Full Paper Submission**

Using the Submit button, submit your journal paper through ICACT Website, then you will get new paper ID of your journal, and send your journal Paper ID to the Secretariat@icact.org for the review and editorial processing. Once you got your Journal paper ID, never submit again! Journal Paper/CRF Template

- **Step 2. Full Paper Review**

Using the evaluation system in the ICACT Website, the editor, reviewer and author can communicate each other for the good quality publication. It may take about 1 month.

- **Step 3. Acceptance Notification**

It officially informs acceptance, revision, or reject of submitted full paper after the full paper review process.

Status	Action
Acceptance	Go to next Step.
Revision	Re-submit Full Paper within 1 month after Revision Notification.
Reject	Drop everything.

- **Step 4. Payment Registration**

So far it's free of charge in case of the journal promotion paper from the registered ICACT conference paper! But you have to regist it, because you need your Journal Paper Registration ID for submission of the final CRF manuscripts in the next step's process. Once you get your Registration ID, send it to Secretariat@icact.org for further process.

- **Step 5. Camera Ready Form (CRF) Manuscripts Submission**

After you have received the confirmation notice from secretariat of ICACT, and then you are allowed to submit the final CRF manuscripts in PDF file form, the full paper and the Copyright Transfer Agreement. Journal Paper Template, Copyright Form Template, BioAbstract Template,

Journal Submission Guide

All the Out-Standing ICACT conference papers have been invited to this "ICACT Transactions on the Advanced Communications Technology" Journal, and also welcome all the authors whose conference paper has been accepted by the ICACT Technical Program Committee, if you could extend new contents at least 30% more than pure content of your conference paper. Journal paper must be followed to ensure full compliance with the IEEE Journal Template Form attached on this page.

➢ How to submit your Journal paper and check the progress?

Step 1. Submit	Using the Submit button, submit your journal paper through ICACT Website, then you will get new paper ID of your journal, and send your journal Paper ID to the Secretariat@icact.org for the review and editorial processing. Once you got your Journal paper ID, never submit again! Using the Update button, you can change any information of journal paper related or upload new full journal paper.
Step 2. Confirm	Secretariat is supposed to confirm all the necessary conditions of your journal paper to make it ready to review. In case of promotion from the conference paper to Journal paper, send us all the .DOC(or Latex) files of your ICACT conference paper and journal paper to evaluate the difference of the pure contents in between at least 30% more to avoid the self replication violation under scrutiny. The pure content does not include any reference list, acknowledgement, Appendix and author biography information.
Step 3. Review	Upon completing the confirmation, it gets started the review process thru the Editor & Reviewer Guideline. Whenever you visit the Author Homepage, you can check the progress status of your paper there from start to end like this, " Confirm OK! -> Gets started the review process -> ...", in the Review Status column. Please don't miss it!

Volume. 4 Issue. 4

Joint iterative channel estimation and decoding under impulsive interference condition

Patcharin Insom*,**, Piyakiat Insom***, Pisit Boonsrimuang***

*Institute of Remote Sensing and Digital Earth, Chinese Academy of Science (CAS), Beijing, China, 100101

** University of Chinese Academy of Science (UCAS), Beijing, China, 100049

***Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand, 10520

insom.patcharin@gmail.com, piyakiat_insom@hotmail.com, kbpisit@kmitl.ac.th

***Abstract*—Even though Low-Density-Parity-Check (LDPC) code which has the decoding performance close to the Shannon Limit and it is designed as a powerful forward-error-correction (FEC) code in the Additive White Gaussian Noise (AWGN) channel, simulation results show that the performance of LDPC decoder is degraded when exposed to the impulsive noise. According to such a impulsive noise impact, joint iterative channel estimation and decoding technique is proposed in this paper so as to decrease the effect of impulsive interference while less complicated in processing. The proposed methods decreases the complexity by implementing the simple way of channel estimation and applying joint iterative technique between channel estimation and LDPC decoding under two kind of impulsive noise; pulsed radio frequency interference(RFI) and symmetric alpha-stable ($S\alpha S$). In the optimal decoder, channel parameter estimation can be as accurate as possible. Because computed in every time of iterative decoder, channel parameters have been always optimized resulting in the enhancement of LDPC decoder performance.**

***Keyword*—LDPC decoding, pulsed RFI, symmetric alpha-stable, Joint iterative, Channel estimation.**

I. INTRODUCTION

Reliability is an important variable in communication systems. Undesirable signal should be eliminated. Not only planetary communications systems, but also satellite communications systems are commonly interrupted by impulsive noise; pulsed radio frequency interference (RFI) and symmetric alpha- stable noise ($S\alpha S$).

Several researches which investigate the impact of pulsed RFI on LDPC decoder performance found that regardless of the level of signal to noise ratio (SNR), the Bit-Error-Rate (BER) is always large [1]. Pulsed RFI frequently has the duty cycles in the order of 5% or smaller through code length. In addition, pulsed RFI randomly turns up in the form of block and the noise variance varies with time. Currently, several applications, such as ISS ACS Transponders, 4th Gen Transponders, Integrated Receivers and satellite communication receiver, are involved with pulsed RFI, resulting in wide awareness of the impact of pulsed RFI.

$S\alpha S$ distribution frequently used for simulating the impulsive interference. $S\alpha S$ noise is constructed based on two important parameters; characteristic exponent and dispersion parameter, in which different value of the two parameters give different heavy tail noise level. $S\alpha S$ noise has impact on several communication system such as Orthogonal Frequency Division Multiplexing (OFDM) [2].

LDPC codes, invented by Gallager, achieve near capacity performance in a wide class of channels [1]. Generally, various simulations and applications are based on AWGN channel. However, our simulation is unable to consider being only AWGN channel because of pulsed RFI and $S\alpha S$ noise appearing. In impulsive channel, an initial Log Likelihood Ratio (LLR) has been developed in many ways [3] [4] [5] [6]. This work constructs the initial LLR based on probability density function (PDF). Recently, joint iterative technique have been extensively implement in several applications [1] [7] [8] and attractive for mitigating the effect of heavy noise. The proposed method presented the joint iterative between channel estimation and LDPC decoding technique which has been developed to re-processing parameter estimation for constructing the PDF during each iterative time of LDPC decoder processes.

On the one hand, pulsed RFI has close form of PDF and the SNR has been investigated by several researchers [4] to obtain the noise variance and LLR in consequently. Although all of them work very well when the SNR value is large, their performance suffers at negative SNR. The iterative SNR estimation [9] is rest on an iterative solution for the maximum likelihood estimation of the amplitude from which the SNR is competent to compute. Its results show that it exhibits a lower bias and normalized mean squared error than other techniques and that the useful range extends to negative SNR.

Manuscript received April 1, 2015. This work was supported self-research scholarship for Master student of Asia-Pacific Space Cooperation Organization (APSCO) and the China Scholarship Council (CSC) under Grant 2011GXZH34.

Patcharin Insom was master student in Master program on Space Technology (MASTA), Beihang University, Beijing, China 100191. She is now PhD. Candidate in the institute of remote sensing and digital earth, Chinese Academy of Science, and the University of Chinese Academy of Science, Beijing, China 100101. (corresponding author to provide phone: +86 131 6156 1499; e-mail: insom.patcharin@gmail.com).

Piyakiat Insom is now an undergraduate student in the department of electronic, Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand, 10520 (e-mail: piyakiat_insom@hotmail.com).

Pisit Boonsrimuang is now a lecturer at telecommunication engineering department, Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand, 10520 (e-mail: kbpisit@kmitl.ac.th).

However, the iterative SNR estimation is time-consuming because of demanding a lot of complicated processes and taking a large of calculating time depending on the number of iteration. On the other hand, there is no close form of PDF for $S\alpha S$ distribution, thus SNR cannot implemented to obtain the noise variance. Geometric signal-to-noise ratio (GSNR) is implemented. Some of research works [10] [11] [12] applied Particle Filter (PF) to estimate the important parameters in $S\alpha S$ distribution and get the appreciated Log Likelihood Ratio (LLR) in consequently. Since the number of particle which deploy in PF process, complexity in processing is concern problem. In this work, we implement commonly use of the second order moment (M_2) method [4] to estimate the noise variance under pulsed RFI channel and use of Logarithmic Moment (LM) [13] which has no complicated mathematic formula to estimate the channel parameter under $S\alpha S$ channel.

The rest of this paper is organized as follows. Section II presents the system model of the two channel type; the pulsed RFI, $S\alpha S$, and the influence on LDPC decoder performance. Section III, which reviews channel estimation techniques, consists of M_2 SNR estimation and GSNR estimation. In section IV, the joint iterative channel estimation and decoding approaches are presented for the both channels. The simulation results of the proposed techniques demonstrate in section V. The conclusion is offered in the last section.

II. System Model

A. Pulsed RFI Model

The system of pulsed RFI channel [1] illustrates as Fig. 1 and (1) where x_k is transmitted sequence and y_k is received sequence. The transmitted signal (x_k) is interrupted by pulsed RFI ($n_{rfi,k}$) and AWGN ($n_{AWGN,k}$) which both of them have zero mean, but possess difference in noise variance. Distribution of AWGN and pulsed RFI are capable of representing as $N(0,\sigma_1^2=\sigma^2+\vartriangle\sigma)$ and $N(0,\sigma_2^2)$ respectively. In practical, because most of the channel is time varying SNR mismatch over an AWGN channel, the noise variance is explained as $\sigma_1^2=\sigma^2+\vartriangle\sigma$ when σ^2 signify as the noise variance of AWGN and $\vartriangle\sigma$ is symbolic of the random walk signal of AWGN.

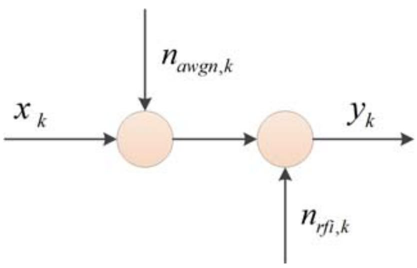


Fig. 1 System model for pulsed RFI experiment.

$$y_k = x_k + n_{awgn,k} + g(t) * n_{rfi,k} \quad (1)$$

Where

$n_{awgn,k} \sim N(0,\sigma_1^2=\sigma^2+\vartriangle\sigma)$

$n_{rfi,k} \sim N(0,\sigma_2^2)$

Fig. 2 shows pulsed RFI in single source. Generally, pulsed RFI randomly turns up throughout the length of x_k and has duty cycles 3-5% of the code length. In addition, pulsed RFI has the noise variance changing with time. Pulse train function ($g(t)$) is assumed to be periodic gating function with pulse repetition rate of $1/T$ and duty cycle τ/T and is used for generating pulsed RFI.

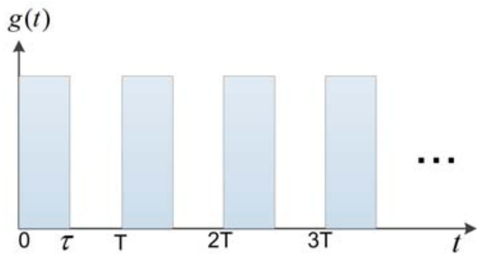


Fig. 2 Gating function for pulsed RFI model.

Fig. 3 demonstrates the influence of pulsed RFI on LDPC decoder performance. It is found that whatever SNR is, BER under pulsed RFI environment is always seriously.

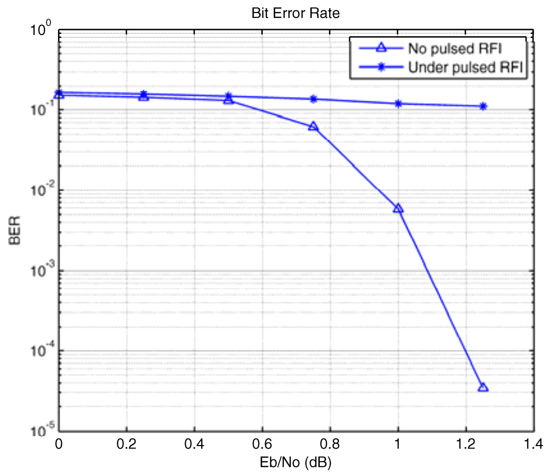


Fig. 3 Pulsed RFI impact on LDPC decoder performance simulated with the AR4JA (8192, 4096) code.

B. Symmetric Alpha-Stable noise Model

The $S\alpha S$ channel describes by a transmitted sequence x_k is interrupted by $S\alpha S$ noise which depict in Fig. 4 and (2), resulting in y_k sequence.

$$y_k = x_k + n_{S\alpha S,k} \quad (2)$$

When we mention to $S\alpha S$ channel, one properties of $S\alpha S$ distribution [14] [15] is that there is no closed form expression of the PDF (except in case of Cauchy and Gaussian distribution), the most convenient way to explain them is using their characteristic function, which define as:

$$\varphi(\omega) = \exp(j\delta\omega - \gamma|\omega|^{\alpha}) \quad (3)$$

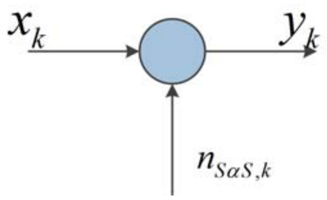


Fig. 4 System model for $S\alpha S$ experiment.

Where

- α is the characteristic exponent. It is vary between $0 < \alpha \le 2$, and it describes the tail of the distribution. A small positive value of α points out the severe impulsiveness, resulting in heavy-tail of the distribution. The distribution is close to Gaussian behavior, if α near to 2. A value of α = 1 means Cauchy distribution.
- γ is the dispersion parameter or scale ($\gamma > 0$). It indicates the spread of the density around the location parameter. It is similar to the variance of Gaussian density. In Gaussian density case, this value is equal to half of the variance of Gaussian density.
- δ is the location parameter ($-\infty < \delta < \infty$). It corresponds to the mean for $1 < \alpha \le 2$, and corresponds to the median for $0 < \alpha \le 1$.

Fig. 5 illustrates the $S\alpha S$ sequence at difference α values. They show different impulsiveness with $\alpha = 1$, $\alpha = 1.2$, $\alpha = 1.7$, and $\alpha = 2$, respectively. An amplitude of impulse for small α cases show very high interrupted level whereas the value of α which is near to 2 illustrates lower amplitude of impulse which no large of interrupting.

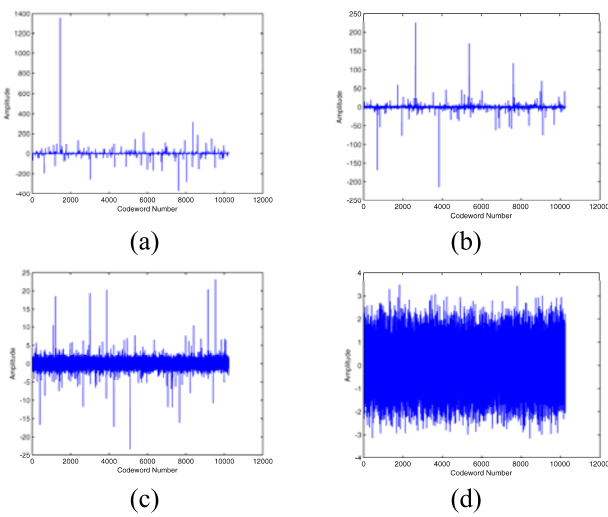


Fig. 5 The received signal interrupting by $S\alpha S$ Noise with (a) $\alpha = 1$; (b) $\alpha = 1.2$; (c) $\alpha = 1.7$; (d) $\alpha = 2$.

When we mention to the alpha-stable distribution, it is a four-parameter family of distributions and is generally denoted by $S(\alpha, \beta, \gamma, \delta)$. In the $S\alpha S$ distribution, β value which explains the skewness of the distribution is equal to 0. We can conclude that the family of alpha-stable distribution is a rich class, and includes the following distributions:

- The Gaussian distribution $N(\mu, \sigma^2)$ is provided by $S(2, 0, \sigma/\sqrt{2}, \mu)$.
- The Cauchy distribution is denoted by $S(1, 0, \gamma, \delta)$.

The PDF of Gaussian and Cauchy distribution are given by:

$$f_{gaussian}(x) = \frac{1}{\sqrt{4\pi\gamma}} \exp\left\{-\frac{(x-\delta)^2}{4\gamma}\right\} \quad (4)$$

$$f_{Cauchy}(x) = \frac{\gamma}{\pi\left[\gamma^2 + (x-\delta)^2\right]} \quad (5)$$

Even though when $0 < \alpha < 2, \alpha \ne 1, 2$, no closed- form expressions exist for the PDF, but we can be computed the PDF, $f(x)$ by taking the inverse Fourier transform of the characteristic function, resulting in:

$$f_{S\alpha S}(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \exp(-\gamma |\omega|^{\alpha}) e^{j\omega(x-\delta)} d\omega \quad (6)$$

Fig.6 shows the impact of SαS noise on LDPC decoder performance when LLR calculate from normal distribution. Transmitted signal is more interrupted when the value of α go to small value.

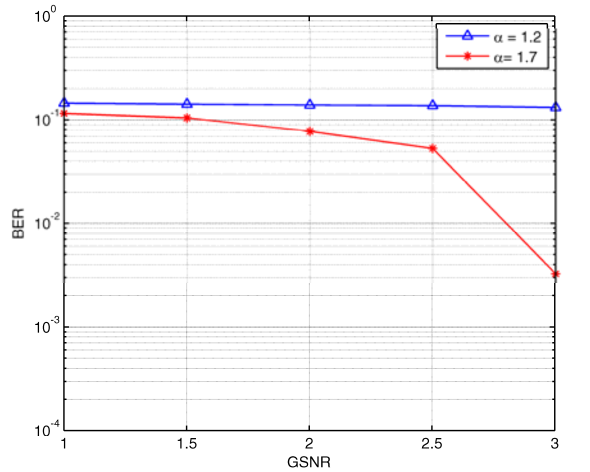


Fig. 6 $S\alpha S$ noise impact on LDPC decoder performance simulated with the AR4JA (8192, 4096) code.

III. Channel Parameters Estimation

A. Pulsed RFI Model

Various digital communication applications, such as power control, bit error estimation, and turbo decoding, involve the knowledge of the SNR. For optimal performance, SNR estimation must be as accurate as possible. Several techniques have been proposed for SNR estimation. M_2 [4] is popular method in several applications because it provides a simple processing. SNR is capable of computing from the ratio of the signal mean squared (A) and the noise variance (σ^2) which both of them are expressed as:

$$A = \frac{1}{N} \sum_{k=1}^{N} |y_k| \quad (7)$$

$$\sigma^2 = \frac{1}{N}\sum_{k=1}^{N}(y_k - A)^2 \tag{8}$$

B. Symmetric Alpha-Stable noise Model

As mentioned above, a difficult task arises in using of $S\alpha S$ model because there is no the closed form of PDF. They do not exist the second-order moment, or the variance. Therefore, another estimated method for noise power is necessary. For instance, in this experiment we use GSNR [15]:

$$GSNR = \frac{1}{2C_g}\left(\frac{A}{S_0}\right)^2 \tag{9}$$

Where
A is the signal amplitude.
S_0 symbolic to the geometric noise power.

$$S_0 = \frac{(C_g\gamma)^{1/\alpha}}{C_g} \tag{10}$$

$C_g = 1.78$ is the experiential of the Euler constant. The constant $2C_g$ in (9) ensures for the SNR in case of $\alpha = 2$ corresponding to the Gaussian distribution with variance $\sigma^2 = 2\gamma$. In digital communication system, we are interested in characteristic of BER in term of E_b / N_0 (the ratio of signal energy per bit to the noise spectral density). This means that the parity bits do not represent transmitted information and their energy must be removed over information bits. Therefore, we can define E_b / N_0 for $S\alpha S$ channel as:

$$\frac{E_b}{N_0} = \frac{GSNR}{2rm} \tag{11}$$

Where r is code rate and m is the number of bits carried per M-array symbol ($m = 1$ for binary code).

IV. Joint Iterative Channel Estimation and Decoding

A. Pulsed RFI Model

The system model which mentioned in the section II points out that both AWGN and pulsed RFI are capable of explaining as normal distribution. Based on the PDF of the distribution, we are able to form the initial LLR of simulated channel. If the assumed noise PDF has distribution as:

$$f(n) = \frac{1-\varepsilon}{\sqrt{2\pi\sigma_1^2}}e^{-\frac{n^2}{2\sigma_1^2}} + \frac{\varepsilon}{\sqrt{2\pi\sigma_2^2}}e^{-\frac{n^2}{2\sigma_2^2}} \tag{12}$$

Where $\sigma_2^2 > \sigma_1^2$

Contaminated Gaussian Log-Likelihood-Ratio (CGLLR) [1] has been constructed. CGLLR provides initial information more accurate than implementing Gaussian LLR. CGLLR is re-written as the follow:

$$CGLLR(y_k) = \log\left(\frac{\frac{1-\varepsilon}{\sqrt{2\pi\sigma_1^2}}e^{-\frac{(y_k-1)^2}{2\sigma_1^2}} + \frac{\varepsilon}{\sqrt{2\pi\sigma_2^2}}e^{-\frac{(y_k-1)^2}{2\sigma_2^2}}}{\frac{1-\varepsilon}{\sqrt{2\pi\sigma_1^2}}e^{-\frac{(y_k+1)^2}{2\sigma_1^2}} + \frac{\varepsilon}{\sqrt{2\pi\sigma_2^2}}e^{-\frac{(y_k+1)^2}{2\sigma_2^2}}}\right) \tag{13}$$

Where

- $\sigma_1^2 = \sigma^2 + \Delta\sigma$ is the noise variance of Gaussian part of the PDF.
- σ_2^2 is the noise variance of the contaminating heavy-tailed PDF (pulsed RFI noise). Chosen ranges are $1.0 < \sigma_2^2 \le 4.0$.
- ε is the percentage of sample from the heavy-tailed PDF. It can range from $0 < \varepsilon \le 0.5$ with the value greater than 0.3 occurring with very low probability.

To implement CGLLR, two parameters must be computed. Firstly, the percentage of sample from the heavy-tailed PDF (ε) or pulsed RFI part is calculated by higher order moments. Fig.7 illustrates detection of pulsed RFI technique by M_2 method. Let N is code length and G is number of group.

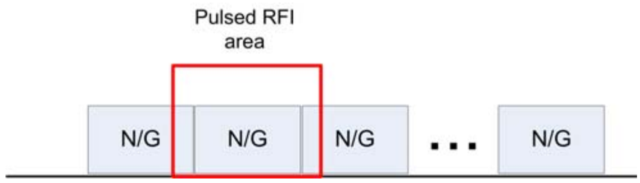


Fig. 7 Detection pulsed RFI by the second order moment.

N is divided into G groups; thus, each group is composed of N/G samples. Because M_2 value of pulsed RFI area is higher than any other region, we take the advantage from this point to capture the area that pulsed RFI appearing. M_2 of every group is calculated and the group which has the largest M_2 value is capable of considering to be pulsed RFI region. Therefore, number of sample of pulsed RFI is approximately equal to N/G. To ensure that all pulsed RFI samples are detected, a few samples both left and right side of that group are included. In other words the pulsed RFI region is approximately equal to the area of group which it has the highest M_2 value including its side-band. Further precision may divide N into several groups and each group contains smaller samples. Applying the same technique; find M_2 of every group, consequently, pulsed RFI area may contain more than one group which have the highest M_2 value respectively. In practice, M_2 are estimated by their respective time averages for both real and complex channels as:

$$M_2 \approx \frac{1}{N}\sum_{k=1}^{N}|y_k|^2 \tag{14}$$

The second factor that needs to compute for implementing CGLLR is the noise variance of Gaussian part of the PDF ($\sigma_1^2 = \sigma^2 + \Delta\sigma$) because the channel status information is unknown. In various applications, M_2 is used because it is the simple way to obtain σ_1^2 and does not take time to calculate. However, the channel in this work is not same as other works since pulsed RFI also appears in the channel. In this environment, M_2 is unable to provide very accurate value. The proposed method to overcome the pulsed RFI can dealing with this problem [1]. The proposed approach is constructed by the principle of joint iterative channel estimation and decoding. It bases on the fundamental that each iterative time of decoding process, the noise variance (σ_1^2) is re-computed by M_2 method to modified information and obtain more accurate noise variance. Schematic of the proposed technique implementing under pulsed RFI shows as Fig. 8 and Algorithm 1.

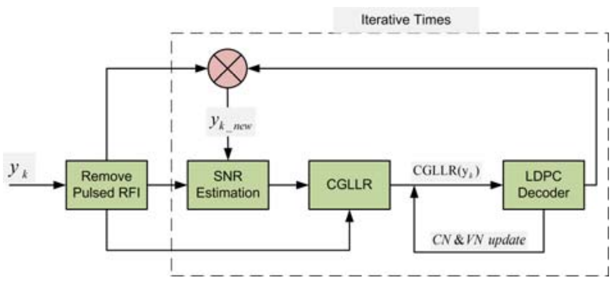


Fig. 8 Schematic block diagram of the proposed technique.

The received signal (y_k) is erased pulsed RFI area which detecting by M_2 method. The received signal after deleting pulsed RFI region denote as y_{k_noRFI}. Consequent step is the noise variance (σ_1^2) evaluation. On the one hand, if it is the first time of LDPC decoder iterations, y_{k_noRFI} will be used to perform the noise variance calculation. On the other hand, the noise variance is computed from y_{k_new} for remaining iteration. $CGLLR(y_k)$ is determined based on the estimated noise variance. We carry out the LDPC decoder only once iterative time. The results consist of hard-decision output, values of check node (CN) , and variable node (VN) updating. Hard-decision outputs are feed back to multiply with the pre- eliminated pulsed RFI signal (y_k) resulting in the new received signal (y_{k_new}). In order to keep LDPC decoder processes, values of check nodes (CN) and variable nodes (VN) updating are also feedback so as to use in the initial state of the next LDPC iteration. In the check stopping criteria step, if the number of iterative time is not equal to the maximum LDPC iterations or decoding error is not equal to zero, go back to step 2.) and increases the number of iterations by one until reach the stopping criterion.

B. *Symmetric Alpha-Stable noise Model*

Joint iteration technique between $S\alpha S$ channel parameter estimation and LDPC decoder allows LLR value updating on fly in every LDPC iterative times, which lead to acquire a high decoder performance. Additionally, because it is not so much mathematics calculation steps in estimated parameter values, the whole processes of the proposed method could has an enhanced results while no computation loss. This section consists of two parts. Firstly, LLR computation is mentioned based on the LM method. Secondly, the joining process of LDPC decoder and estimation of $S\alpha S$ channel parameters is described.

Algorithm 1 Joint iterative between channel estimation and decoding under pulsed RFI condition.

1: **procedure** pulsedRFI (y_k)
2: **Eliminate pulsed RFI**
3: **if** decoding iterative =1 **then**
$\sigma_1^2 = \mathrm{var}(y_{k\_noRFI})$
4: **else**
$\sigma_1^2 = \mathrm{var}(y_{k\_new})$
5: **end if**
6: **LLR computation**
7: **Once iterative time LDPC decoder**
8: **Output feedback**
$y_{k\_new} = y_k \cdot hard - decision$
Check Node (CN) update vale
Variable Node (VN) update value
9: **Check stopping criteria**
10: **if** iterative time = max iteration || Error=0 **then**
Go to 13
11: **else**
Back to 3
12: **end if**
13: **Output codeword**
14: **end procedure**

In (3) there are two key parameters for calculate the PDF of $S\alpha S$ distribution: characteristic exponent (α) and dispersion parameter (γ). The location parameter (δ) can be discarded because it is a position and no influence on the simulation analyses. In this experiment, those two parameters are computed from LM method [13]. Let $X \sim S(\alpha, 0, \gamma, 0)$ is sequence of $S\alpha S$ signal then:

$$L_1 = \mathrm{E}\left[\log|X|\right] = \psi_0(1-\frac{1}{\alpha}) + \frac{1}{\alpha}\log|\frac{\gamma}{\cos\theta}| \tag{15}$$

$$L_2 = \mathrm{E}\left[(\log|X| - \mathrm{E}\left[\log|X|\right])^2\right] = \psi_1(\frac{1}{2}+\frac{1}{\alpha^2}) - \frac{\theta^2}{\alpha^2} \tag{16}$$

Where
L_1 is logarithmic moment at order 1
L_2 is logarithmic moment at order 2

$$\psi_0 = -0.57721566..., \qquad \psi_1 = \frac{\pi^2}{6} \tag{17}$$

We can solve for α by setting $\theta = 0$ and then calculate for γ value. The LLR calculation for $S\alpha S$ channel is related to the PDF of the $S\alpha S$ distribution which express as:

$$LLR(y_k | x_k) = \ln\left(\frac{P(y_k | x_k = +1)}{P(y_k | x_k = -1)}\right) \tag{18}$$

In $S\alpha S$ channel, the joint iterative between channel estimation and LDPC decoder perform as Fig. 9 and Algorithm 2. The processes begin with an initial LLR value.

For the first iteration of LDPC, LLR is calculated by setting α equal to 2 (Gaussian LLR). According to (15) - (16), we can obtain the γ value. From these two parameter values, we can compute LLR. The initial LLR value are input to LDPC decoder resulting in three output parameters.

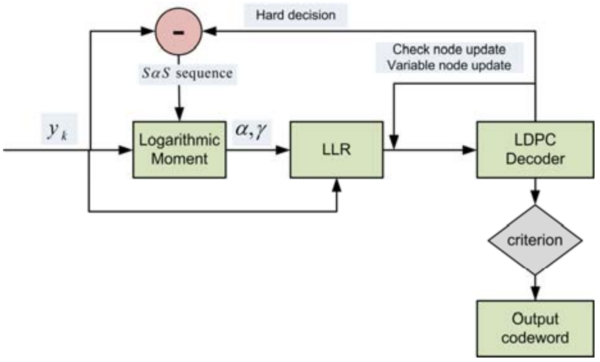


Fig. 9 Schematic block diagram of the proposed technique for joint iterative between channel estimation and decoding under $S\alpha S$ channel.

The first two outputs are variable node and check node values of LDPC, which will be used for the next iteration of the decoder. The third output is hard-decision, it will be feedback to subtract from received signal (y_k) to estimate α and γ values by LM method before calculating the LLR value for the next loop of LDPC decoder. The output from subtracting hard-decision with received signal (y_k) is $S\alpha S$ noise sequence, which can be used to estimate α and γ values as (15)-(16). Because two important parameters are estimated in every iterative time, the LLR value can update to more accurate value which has influence on the decoder performance. The loops are continuing until reach the criterion and get output codeword in the final step.

V. Simulation Results and Discussion

The LDPC code studied here is the AR4JA (8192, 4096) code recommended by the Consultative Committee for Space Data Systems (CCSDS) and it has been chosen for some deep space applications. In this section, BER of the joint iterative channel estimation and decoding or the proposed approaches are presented for both under pulsed RFI and $S\alpha S$ channel.

A. Pulsed RFI Model

In the simulation, we set the noise variance of the contaminating heavy-tailed (pulsed RFI) PDF (σ_2^2) by 4.0 and set the random walk signal of AWGN ($\Delta\sigma$) by 10% of the noise variance of AWGN part (σ_1^2). The proposed method was compared the result with the conventional method which performs noise variance of AWGN part (σ_1^2) estimation by M_2 SNR estimation based on hard-decisions of the previous codeword which they are feedback to multiply with that codeword to eliminate modulated information. Fig. 10 depicts that the proposed technique has only about 0.2 dB at BER = 10^{-3} different from the SNR know experiment. The efficiency of the proposed approach is close to the performance of ideal technique because the noise variance (σ_1^2) is re-calculated in every time of iterative LDPC decoder.

Algorithm 2 Joint iterative between channel estimation and decoding under $S\alpha S$ noise.

1: **procedure** SaSnoise (y_k)
2: **LLR computation**(α,γ,y_k)
3: **if** decoding iterative =1 **then**
$\alpha = 2$
$\gamma = LM(\alpha, S\alpha S$ sequence)
4: **else**
$\alpha = LM(S\alpha S$ sequence)
$\gamma = LM(\alpha, S\alpha S$ sequence)
5: **end if**
6: **Once iterative time LDPC decoder**
7: **Output feedback**
$S\alpha S$ sequence $= y_k - hard\_decision$
Check Node (CN) update vale
Variable Node (VN) update value
8: **Check stopping criteria**
9: **if** iterative time = max iteration || Error=0 **then**
Go to 12
10: **else**
Back to 2
11: **end if**
12: **Output codeword**
13: **end procedure**

Therefore, the noise variance (σ_1^2) has always been changed and obtains more accurate value in each time of LDPC iteration.

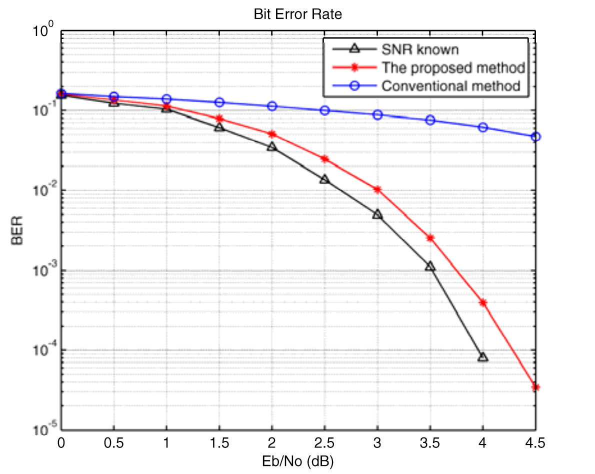


Fig. 10 Bit-Error-Rate performance in the jiggling AWGN channel simulated with the AR4JA (8192, 4096) code.

Table I. makes a comparison of the single block information for SNR estimation of those two techniques. Two kind of operation are taken into our consideration; addition, and multiplication.

TABLE I
COMPARISON OF SINGLE BLOCK INFORMATION FOR SNR ESTIMATION OF THE TWO METHOD

	Conventional method	Proposed method
Addition	$M-1$	$(M-1)k$
Multiplication	$M+1$	$(M+1)k$

Let length of single block information which pulsed RFI is removed is M . k represents the number of LDPC iteration.

In the single block information, the conventional method requires the smallest operational numbers. The proposed approach has a little more computational complexity than the conventional technique depending on the number of LDPC iteration (k).

B. Symmetric Alpha-Stable noise Model

There are two test cases of α value in this simulation: $\alpha = 1.2$ and $\alpha = 1.7$ which is no closed form PDF for these α values. The results demonstrate the proposed method for the joint iterative method between $S\alpha S$ channel estimation and LDPC decoder in term of BER as depict in Fig. 11. BER curve of $\alpha = 1.2$ is more corrupt than BER curve of $\alpha = 1.7$. This is because $\alpha = 1.2$ generate stronger impulsiveness to the channel, thus, it needs high level of GSNR value to obtain a satisfied performance. In the curves, GSNR equals to 2.85 could give BER about 10^{-4} in case of $\alpha = 1.2$ and the GSNR equals to 2 could get BER about 10^{-4} when $\alpha = 1.7$, because $\alpha = 1.7$ is near to be Gaussian noise and has smaller impulsiveness on channel than $\alpha = 1.2$.

From Fig 6 and 11, it is clear that the proposed method has ability to mitigate the effect of $S\alpha S$ noise better than the conventional method which widely uses normal distribution to calculate an initial LLR for LDPC decoder.

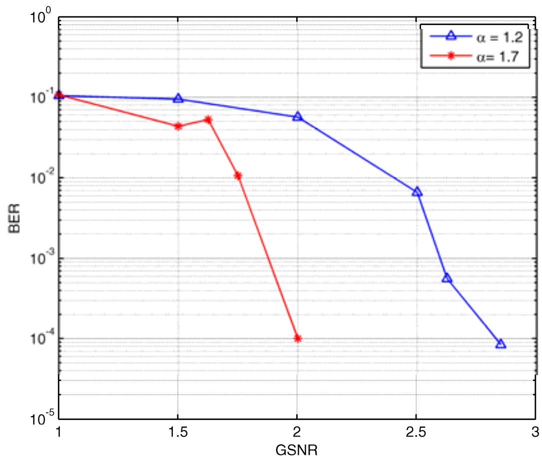


Fig. 11 Bit-Error-Rate performance in the $S\alpha S$ channel when $\alpha = 1.2$ and $\alpha = 1.7$ simulated with the AR4JA (8192, 4096) code.

TABLE II
THE ESTIMATED PARAMETER VALUES OF $S\alpha S$ CHANNEL IN EVERY LDPC ITERATIVE TIME

	Characteristic exponent (α)	Dispersion parameter (γ)
Iteration =1	2.0000	1.2377
Iteration =2	1.7153	0.3491
Iteration =3	1.7304	0.3289
Iteration =4	1.7551	0.3218
Iteration =5	1.7425	0.3205
.	.	.
.	.	.
.	.	.
Iteration =49	1.7099	0.3384
Iteration =50	1.7099	0.3384

Table II demonstrates an updating of two important parameters of SαS channel in the case of GSNR = 2 in fifty iterative times. Both α and γ tend to go to the true value in every iterative time, which established by α = 1.7 and γ = 0.5338. Although, the updating of γ parameter is unable to reach the true value, it is better than implement LDPC decoder with LLR from normal distribution, which both two parameters have the values same as setting in the first iteration time. The updated parameter reflects the enhancement of LLR calculation that mean LLR value can go to the appreciated values which has influence to decoder performance.

VI. CONCLUSION

Joint iterative between channel estimation and decoding have been presented in this work for pulsed RFI and $S\alpha S$ channel. The second order moment method have been implement for estimating the noise variance in pulsed RFI channel and logarithmic moment is used for estimating the important two parameter under $S\alpha S$ channel. In addition, a joint iterative technique also provides a chance to update channel parameters in every iterative time of the decoder which is useful for decoder performance. The results demonstrate that the proposed technique can mitigate the impact of the impulsive noise on LDPC decoder for both pulsed RFI and $S\alpha S$ channel. The proposed method is attractive to apply to various applications which pulsed RFI or $S\alpha S$ is involved with their requirements.

ACKNOWLEDGMENT

The authors are thankful to the reviewers for their detailed reviews and constructive comments and suggestions, which greatly improve the quality of this paper.

REFERENCES

[1] P. Insom, R. Liu, R. Duan, Y. Hou and P. Boonsrimuang, "Joint iterative channel estimation and decoding under pulsed radio frequency interference condition," in *IEEE 16th Int. conf. Adv. Commun. Technol.(ICACT).*, 2014.

[2] N. Jaoua, E. Duflos, P. Vanheeghe, L. Clavier and F. Septier, "Impulsive interference mitigation in ad hoc networks based on alpha-stable modeling and particle filtering," in *IEEE Int. conf.Acoustics, Speech and Signal Process (ICASSP).*, 2011.

[3] T. S. Shehata, I. Marsland and M. El-Tanany, "Near Optimal Viterbi Decoders for Convolutional Codes in Symmetric Alpha-Stable Noise," in *IEEE 72nd Veh.Technol. Conf. Fall (VTC 2010-Fall).*, 2010.

[4] D. R. Pauluzzi and N. C. Beaulieu, "A comparison of SNR estimation techniques for the AWGN channel," *IEEE Trans. Commun.*, vol. 48, no. 10, pp. 1681-1691, 2000.

[5] T. S. Saleh, I. Marsland and M. El-Tanany, "Low-complexity near-optimal map decoder for convolutional codes in symmetric alpha-stable noise," in *IEEE 25th Canadian Conf. Elect. and Comput. Eng. (CCECE).*, 2012.

[6] T. S. Saleh, I. Marsland and M. El-Tanany, "Simplified LLR-based Viterbi decoder for convolutional codes in symmetric alpha-stable noise," in *IEEE 25th Canadian Conf. Elect. and Comput. Eng.(CCECE).*, 2012.

[7] H. D. Pfister and K. Byung-Hak, "An Iterative Joint Linear-Programming Decoding of LDPC Codes and Finite-State Channels," in *IEEE Int. conf. Commun.*, 2011.

[8] J. Chen, F. Yang, S. Zhang, L. Tang, L. Luo and L.-N. Tho, "Joint iterative decoding for irregular-LDPC-based coded cooperation over a non-ideal relay channel," in *IEEE Int. conf. Wireless Commun and Signal Process (WCSP).*, 2009.

[9] L. Bin, R. DiFazio and A. Zeira, "A low bias algorithm to estimate negative SNRs in an AWGN channel," *IEEE Commun. Lett.*, vol. 6, no. 11, pp. 469-471, 2002.

[10] W. Shuang, C. Lijuan, S. Cheng, Z. Yan, M. Yeary and W. Qiang,

"Noise Adaptive LDPC Decoding Using Particle Filtering," *IEEE Trans. Commun.,* vol. 59, no. 4, pp. 913-916, 2011.

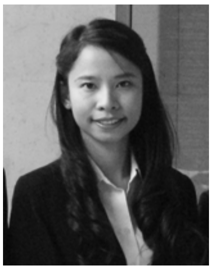
[11] C. Lijuan, W. Shuang, S. Cheng and W. Qiang, "Noise adaptive LDPC decoding using particle filter," in *IEEE 43rd Annu. Conf.Inform. Sci. and Syst.(CISS).*, 2009.

[12] L. Mihaylova, P. Brasnett, A. Achim, D. Bull and N. Canagarajah, "Particle filtering with alpha-stable distributions," in *IEEE 13th Workshop Statist. Signal Process.*, 2005.

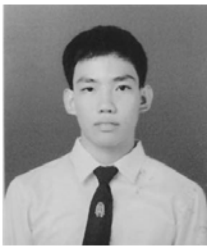
[13] E. E. Kuruoglu, "Density parameter estimation of skewed alpha-stable distributions," *IEEE Trans.Signal Process.,* vol. 49, no. 10, pp. 2192-2201, 2001.

[14] M. Shafieipour, H.-S. Lim and T.-C. Chuah, "Decoding of Turbo Codes in Symmetric Alpha-Stable Noise," *ISRN Signal Process.,* vol. 2011, p. 7, 2011.

[15] T. C. Chuah, "Robust iterative decoding of turbo codes in heavy-tailed noise," *IEE Proc. Commun.,* vol. 152, no. 1, pp. 29-38, 2005.



Patcharin Insom received the B.E. degree in Telecommunication Engineering (Second Class Honors), 2009 from King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand. In 2013, she received the M.E degree in space technology application from Beihang University, Beijing, China supported by APSCO and CSC scholarship. She currently is Ph.D candidate at the University of Chinese Academy of Sciences, Beijing, China sponsor by CAS-TWAS president fellowship. Her interested research is wireless communication and remote sensing applications.



Piyakiat Insom currently is bachelor degree student in Electronic Engineering, Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang, and Bangkok, Thailand. His interested researches are included micro-controller and wireless communication.

Pisit Boonsrimuang received the B.E. and M.E. degrees in telecommunication engineering from King Mongkuts Insti- tute of Technology Ladkrabang (KMITL), Thailand and Dr.E. from Mie University, Japan, in 1997, 2000 and 2007 respec- tively. He is currently working as lecturer at King Mongkuts Institute of Technology Ladkrabang, Thailand. His research inter- ests include transmission techniques for future multimedia wireless LAN system.

A Model for Network Traffic Anomaly Detection

Nguyen Ha Duong*, Hoang Dang Hai**

*Faculty of Information and Technology, National University of Civil Engineering, 55 Giai Phong, Hanoi, Vietnam

** Posts and Telecommunications Institute of Technology, Ministry of Information and Communication, Nguyen Trai, Hanoi, Vietnam

duongnh@nuce.edu.vn, hdhai@mic.gov.vn

***Abstract*—Network traffic anomaly detection can find unusual events cause by hacker activity. Most research in this area focus on supervised and unsupervised model. In this work, we proposed a semi-supervised model based on combination of Mahalanobis distance and principal component analysis for network traffic anomaly detection. We also experiment clustering technique with suitable features to remove noise in training data along with some enhanced detection technique. With the approach of combining anomaly detection and signature-based detection system, we believe the quality of normal dataset will greatly improve.**

***Keyword*—Network traffic anomaly, anomaly detection, semi-supervised model, intrusion detection, network security**

I. INTRODUCTION

TODAY, network security is world-wide major concern of many countries. Organizations, companies and agencies are often facing with network attacks. The Intrusion Detection System (IDS) is implemented as an effective device to detect attacks outside or inside of a network. However, typical IDS often rely on signature database or pattern of known attack [1][2]. Therefore, intruders can change some parameters or characters that different from known patterns to make IDS unable to detect the new variances. Anomaly detection is the approach of recent IDS [3-6], since it does not require any prior knowledge about the attack signatures. Thus, it is capable to detect new attacks. Anomaly detection system (ADS) is used to detect the abnormal behaviour of a system. ADS can operate independently or as a component of IDS.

There are various network anomaly detection methods in recent years including machine learning techniques, statistical-based methods, principal component analysis (PCA) methods, etc. A review of different approaches for anomaly detection was given in [1-7]. Various methods and techniques proposed for anomaly detection indicate the difficulties of network traffic anomaly detection. There are several reasons: 1) the techniques of attackers become more sophisticated. There are many types of anomalies with different traffic data features. 2) No existing method is considered better than the others due to the complexity of the anomalies. Several issues remain unsolved regarding detection speed, accuracy, confidence, complexity, etc. Among various anomaly detection approaches, PCA has been proposed as an effective solution [8-10]. PCA is useful to reduce the complexity of the dataset while maintaining significant dataset features. From high level viewpoint, anomaly detection can be categorized into 3 models [7]:

1) Supervised: models both normality and abnormality. The entire area outside the normal class represents the outlier class. Supervised detection techniques fail to recognize behaviour that is not previously modelled, thus it lacks the ability to classify unknown anomalies.
2) Unsupervised: no prior knowledge of the data is needed. It processes the data as a static distribution, pinpoints the most remote points, flags them as potential outliers (anomaly).
3) Semi-Supervised: models only normality. It needs pre-classified data but only learns data marked normal. It is suitable for static or dynamic data as it only learns one class which provides the model of normality (baseline). If a point’s distance exceeds the established threshold from the normal baseline, it is considered abnormal point.

In this paper, we propose a semi-supervised model using a modified Mahalanobis distance based on PCA (M-PCA) for network traffic anomaly detection. In order to reduce the noise of anomalies, we propose to use the K-means clustering algorithm to group similar data points and to build normal profile of traffic. This algorithm helps to improve the quality of the training dataset. The remainder of the paper is organized as follows: Section 2 presents related previous works. Section 3 proposes our research model. Section 4

Manuscript received April 30, 2015. This work is a follow up of the accepted conference paper as an outstanding paper for the 17th International Conference on Advanced Communication Technology.

Nguyen Ha Duong is with the Faculty of Information and Technology, National University of Civil Engineering (corresponding author to provide phone: +84-98-756-7271; fax: +84-04-3869-1910; e-mail: duongnh@ nuce.edu.vn).

Hoang Dang Hai was with Vietnam Computer Emergency Response Teams (VNCERT), Vietnam. He is now with the Posts and Telecommunications Institute of Technology, Ministry of Information and Communication, Vietnam (e-mail: hdhai@mic.gov.vn).

proceeds with our experiment and results. Then, concluding remarks are provided in Section 5.

II. RELATED WORKS

The authors in [3-7] presented a review of anomaly based intrusion detection systems. A version of apriori algorithm was used with systolic arrays to build efficient pattern matching similar to a signature based method. In [5], the existence of irrelevant and redundant features has been studied that affect the performance of machine learning part of the detection system. This work showed that a good selection of the features will result in better classification performance. The authors in [6] demonstrated that the elimination of the unimportant features and irrelevant features did not reduce the performance of the detection systems.

Anomaly detection models based on PCA was proposed by Shyu in [8]. PCA was used together with outlier detection in assumption that the anomalies appear as outliers to the normal data. PCA can reduce the dimensionality of the dataset. The authors in [7-10] further improved PCA algorithm in combination with several algorithm such as sketch-based and signal-analysis based in a framework. The authors in [11] showed that an important advantage of combining redundant and complementary classifiers is to increase accuracy and better overall generalization. Several authors [9-12] also identified important input features in building IDS that are computationally efficient and effective. This work showed the performance of various feature selection algorithms including bayesian networks, classification and regression trees.

Several works provided experiments using KDD-CUP'99 dataset [13-16], which is a subset of the Intrusion Detection Evaluation dataset of DAPRA. Almost works proposed to use all features from the raw data of this dataset.

III. PROPOSED MODEL

A. PCA and Mahalanobis Distance

Principal Component Analysis (PCA) is a method for identifying patterns in data, and expressing the data in such a way as to highlight their similarities and differences [8]. PCA produces a set of principal components (PCs), based on eigenvalue/eigenvector pairs. Eigenvalues/eigenvectors can be built from covariance or correlation matrix

The covariance or correlation of any pair of PCs is equal to zero. PCA produces a set of independent variables so the total variance of a sample is the sum of all the variances accounted for by the PCs.

Outlier detection techniques are used to calculate the distance of captured live network data to the normal data projected by PCA procedure. We proposed using Mahalanobis distance for outlier detection, thus outliers measured are presumably anomalous network connections. Any network connection with a distance greater than an established threshold value is considered an outlier. The equation [8] of Mahalanobis distance d between observation x and the sample mean μ is:

$$d^2(x,\mu) = (x-\mu)'S^{-1}(x-\mu) \quad (1)$$

where: S^{-1} is the sample covariance matrix.

(1) takes into account the covariance matrix, thus, it can measure correlation between variables. In this paper, we use correlation matrix instead of covariance matrix since many variables in the training dataset were measured on different scales and ranges. The drawback of this method is the computationally demanding when calculating the inverse of the correlation matrix for feature vectors with a large number of dimensions. We need a method to calculate this distance more efficiently for each new connection. As in [8], the sum square of standardized PCs score is equivalent to the Mahalanobis distance of the observation x from the mean of the sample as follows:

$$d(x,\mu) = \sum_{i=1}^{p} \frac{y_i^2}{\lambda_i} \quad (2)$$

where: y_i is i^{th} the PC score, λ_i is the i^{th} eigenvalue, μ is the mean vector of the trained data set.

(2) is not the best choice equation for outlier detection. Some outlier in y_i^2/λ_i can be small and the total sum of all y_i^2/λ_i take little account for that outlier. One advantage of PCA is the PCs can be sorted in order of decrease eigenvalue $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_p > 0$ then using only some PCs to calculate distance and find outlier. The number of retained PCs is the weight (W) in our work. We need experiment and choose the effective W for anomaly detection.

Intuitively, we assume that the last PCs (*minor PCs*) contain variances which are inconsistent with the data structure of the original variables as indication of outlier. In experiment, we found that using only the *minor PCs* (1-threshold method) with the weight W_2 can achieve good detection result. Any observation has distance greater than an

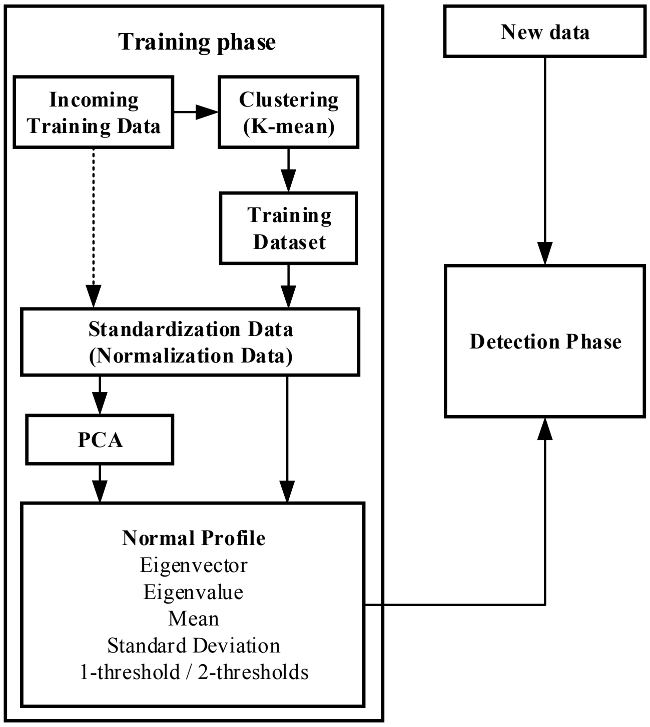


Fig. 1. The proposed model for anomaly detection.

established threshold is considered abnormal event or attack.

Other method is using 2 set PCs: *major PCs* (the most significant PCs) with the weight W_1 and *minor PCs* (least significant PCs) with the weight W_2. Each set has separate distance calculation and upper threshold. Any observation has distance greater than corresponding threshold is considered outlier. *Major PCs* often capture normal trend in variance of original variables. The use of both *major* and *minor PCs* is called 2-thresholds method. We call both 1-threshold and 2-thresholds methods with a common name: M-PCA method.

B. The Basic Model

Fig. 1 shows our proposed model for anomaly detection using M-PCA method and a method to reduce the noise in training data.

The core principle of anomaly detection is calculating the distance and building normal traffic profile. Distance describes how far a point compared to a centre of the known distribution. Our anomaly detection scheme require 2 phases: Training phase and detection phase.

C. Training Phase

The purpose of training phase is building normal traffic profile from normal data pattern. It has following steps:

Step 1: Choose the features $X_1, X_2, \dots, X_p$ which affect normal profile (p is the number of the features used in training and detection phase). Build the normal profile on selected features will reduce the number of dimensions needed to process. PCA is used to analyse the contribution of each feature to PCs of normal data.

Step 2: The network traffic needs to be free of attacks at training time in order to get a snapshot of captured network traffic for training dataset. In reality, this traffic can contain some small attacks considered as noise. Thus, we need to clean it beforehand. We propose using K-means clustering algorithm to remove outliers (noise) of the input data. We assume that the noise is much lesser than the normal data and recommend accepted noise level approximately at 10% of all incoming training data.

Step 3: The cleaned training data needs to standardize:

$$z_k = \frac{x_k - \overline{x}_k}{\sqrt{s_k}} \tag{3}$$

where: $\overline{x_k}$ and s_k is the sample mean and sample variance of the feature X_k in trained data set respectively; z is the standardized vector of training data set, and $z = (z_1, z_2, \dots, z_p)'$

Step 4: Calculate the correlation matrix then pairs of eigenvector and eigenvalue.

Step 5: Compute the PC score of each sample in training data with z and eigenvector:

$$y_i = e'_i z \tag{4}$$

where: y_i is i^{th} the PC score, e_i is the i^{th} eigenvector

Step 6: Sort the PCs by eigenvalues in descending order: $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_p > 0$

Step 7: Compute distance for each observation of training dataset with 1-threshold or 2-thresholds method. 1-threshold method use only minor PCs while 2-thresholds method use both major PCs and minor PCs:

$$d(major\ PCs) = \sum_{i=1}^{q} \frac{y_i^2}{\lambda_i} \tag{5}$$

$$d(minor\ PCs) = \sum_{i=r}^{p} \frac{y_i^2}{\lambda_i} \tag{6}$$

where: $0 < q < r < p$

Step 8: Build the empirical cumulative distribution function (ECDF) of distances. Choose the thresholds corresponding to estimated false positive ratio.

D. Remove noise of input data by K-means

K-means clustering help to clean the data when it have more noise than it should be. The noise can be recognized as attack and normal connection which are outlier with other normal connection. K-means is a clustering analysis algorithm that groups objects based on their feature values into K disjoint clusters. Objects classified into the same cluster have similar feature values. K is a positive integer number specifying the number of clusters, it has to be given in advance. Here are the steps of the K-means clustering algorithm:

Step 1: Define the number of clusters K and initialize K cluster centroids. This can be done by arbitrarily dividing all objects into K clusters, computing their centroids, and verifying that all centroids are different from each other. The centroids can be initialized to various objects chosen arbitrarily.

Step 2: Iterate over all objects and compute the distances to the centroids of all clusters. Assign each object to the cluster with the nearest centroid.

Step 3: Recalculate the centroids of both modified clusters.

Step 4: Repeat step 2 until the centroids do not change any more.

The distance we used in K-means algorithm is Pearson correlation distance [15]. Pearson correlation measures the similarity in shape between two profiles. The formula for the Pearson Correlation distance is:

$$d = 1 - c \tag{7}$$

where: $c = z(u) \cdot z(v)/n$ is the dot product of the z-scores of the vectors u and v. The z-score of u is constructed by subtracting from u its mean and dividing by its standard deviation. Each centroid is the component-wise mean of the points in that cluster, after centering and normalizing those points to zero mean and unit standard deviation.

E. Detection Phase

In detection phase we use the sub-score of M-PCA method to detect each new point from the distribution of the trained data point.

This phase match each new observation with established normal profile to detect anomaly. This include following steps:

Step 1: Standardize data with means and variances from sample training dataset.

Step 2: Compute PC score of each observation with trained eigenvectors which map observed data to subspace.

Step 3: Compute distances of each observation as in (5), (6). A new connection will have 1 or 2 distance values depend on 1-threshold method or 2-thresholds method.

Step 4: Compare thresholds and detection decision: If new connection's distance is greater than any of the established threshold, it marks as anomaly connection. Otherwise, it is normal connection.

1-threshold method:

If $d(minor\ PCs) > d_2$, classify new connection as abnormal

Else $d(minor\ PCs) \leq d_2$ classify new connection as normal

2-thresholds method:

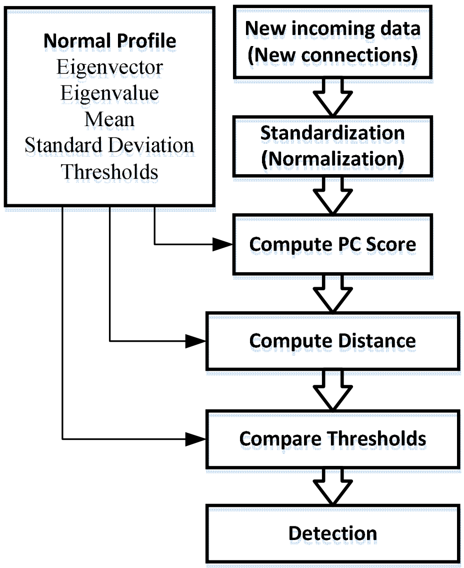


Fig. 2. Detection phase of the model.

If *d(major PCs)* > d_1 OR *d(minor PCs)* > d_2, classify new connection as abnormal

Else *d(major PCs)* ≤ d_1 AND *d(minor PCs)* ≤ d_2 classify new connection as normal

Where: d_1 and d_2 are thresholds of *major PCs* and *minor PCs* respectively.

F. Proposal Enhancement of the Model

In our approach, Anomaly Detection System (ADS) works as an inherent component with signature-based detection system (other name is misuse IDS). ADS alone is a system of suspicious events detection. ADS will collect suspicious events and these events will be validated by signature database, administrator (human) (or supervised classification modules). Misuse IDS can detect intrusion base on packets while ADS often work with connections or flows which limit its response time. For this reason, ADS appropriate for the role of informer for network signature-based IDS generally. In this way, all components of detection system work in circle of spy and detective network manner.

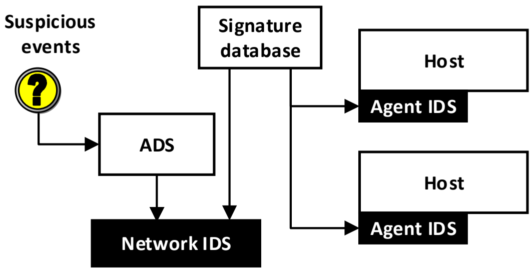


Fig. 3. ADS work as informer for network signature-based IDS

Misuse system rely on signature which have many rules and need to update regularly. Some IDSes work as agents (sensors) for hosts (computers) which only detect intrusion related to those hosts. Using all rules with agent-IDSes will overload the resource available in these hosts. Normally, only a subset of rules are active for some specific services in each host. If we use ADS to check anomaly traffic and then validate by a network signature-based IDS, the detection system will concentrate on more important events from ADS. That's why the precision of ADS system is important because too many false alarm will make ADS becomes unreliable. The number of output alarm from ADS can be large. There should be a high performance network signature-based IDS to validate the result from ADS frequently. Validation can happen at packet, content, connection, flow level. Misuse detection cannot detect unknown attack types. In theory, ADS can detect novel attacks and then validate by human or supervised machine learning. From post-processing, new signature can be generated.

Next section describe an enhancement model which integrate our ADS with misuse IDS. The model focus on the training phase because the quality of normal dataset is very important. Fig. 4 shows the enhancement of the model.

At first, incoming training data will go directly to training dataset pool if the administrator can guarantee there are no attacks. Otherwise, if incoming data contains some noise (attacks), clustering module (K-means or other algorithm) will be used to filter noise. For filter noise at more depth, input training data can go through the signature-based detection module to check and remove known attacks. All rule in signature database must be used because using only subset of rules may let some known attack connections pass as negative. Connection pass all the signature-checking as negative will be considered data for clustering. However, clustering rely on selection of some specific features or variables. In case that feature set is not available, clustering module need to disable and data pass signature detection can be added to training dataset before PCA step.

New data after pass of the anomaly detection module will go through validation checking. This step validates the correct of detection result and identify attack types. If the system only detect and cannot classify type of abnormal events, the detection result is almost nonsense. Signature database is used again to automatic check for attack types with all positive records from anomaly module. Normal data is often has good detection rate and outnumber attack data. For negative records, to avoid overwhelming with large number of data, signature database only need to check randomly chosen records. After validation, these normal records use as feedback to the normal dataset. This way, system doesn't waste resource for many unnecessary negative records and saves resource for identify attacks in positive records.

If anomaly module has too many false alarm records which discover by signature module, all records have to be checked by signature module whether they are positive or negative. And administrator must check the anomaly module, compare score with result from signature module to find what is wrong.

Positive records cannot identified with signature module can go through manual check. Identify attacks by human is challenged task and should only implement for novel attacks.

If the system already built pre-classified attack classes in machine learning database, positive records can be checked with this database to classify attack types. In this manner, system works as supervised model.

Data in training set need to have aging time. New cleaned training data will replace oldest data. Some old attack records still in the normal dataset will be gradually removed. This will help regulate the training data more efficiently.

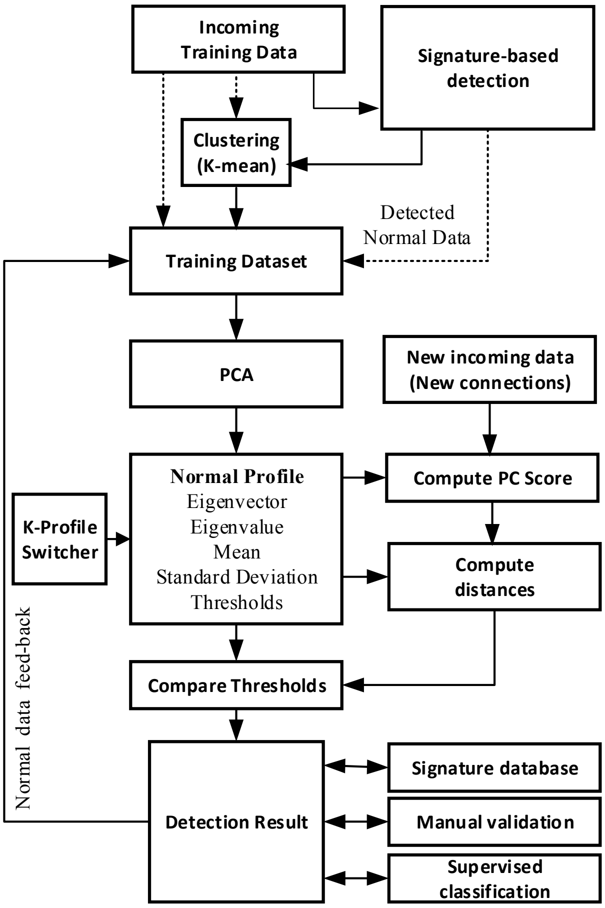


Fig. 4. The Enhancement model

Profile of network can change due to some reasons. Administrators build multiple profiles based on daily usage of network traffic characteristics. A profile switcher can be programmed to change suitable profile accordingly.

IV. EXPERIMENT AND RESULTS

A. NSL-KDD Dataset

In [8], Shyu's test results use KDD CUP 99, the old data set contains 75% of redundant records. KDD CUP'99 is the mostly widely used data set for anomaly detection. The inherent problem of KDD dataset leads to new version of NSL KDD dataset that are mentioned in [14-16]. In [16], the authors conducted a statistical analysis on this data set and found two important issues which highly affect the performance of evaluated system, and results in very poor evaluation of anomaly detection approaches. To solve these issues, they proposed a new data set, NSL-KDD, which consists of selected records of the complete KDD data set [14] and does not suffer from any of the mentioned shortcomings.

We don't use the KDDCUP'99 but the NSL-KDD instead, since the advantages of NSL KDD dataset are as follows:

1) No redundant records in the train set, so the classifier will not produce any biased result.
2) No duplicated records in the test set which have better reduction rates.
3) The number of selected records from each difficult level group is inversely proportional to the percentage of records in the original KDD data set.

The NSL-KDD data includes 41 features and 5 classes that are normal and 4 categories of attacks: Denial of Service Attack (DoS), Probe, Remote to Local Attack (R2L), and User to Root Attack (U2R). Attack categories and types in NSL-KDD data set were given in [14].

B. Performance Measures

We use the following performance measures:
True Positive (TP): the event when an attack connection correctly detected.
True Negative (TN): the event when a normal connection correctly detected.
False Positive (FP): the event when a normal connection detected falsely as attack connection.
False Negative (FN): the event when an attack connection detected falsely as normal connection.
Precision: The ratio of true positive and the number of detected connection as attack.
True Positive Rate (Recall): The ratio of true positive and the number of real attack in the sample data set.
False Positive Rate (FPR): The ratio of false positive and the real number of normal connection in sample data set.
Total Accuracy: The overall successful prediction of both attack and normal connection.

C. Removing Noise by K-means

We implemented experiments in Matlab R2013a. We use the KDDTrain+ dataset [14] for both training and detection phase.

Through experiments we found some features in NSL-KDD are significantly affects the normal profile (Table I). This will make the processing data faster. The K-means clustering algorithm was used with input training data to derive a more cleaned data. The goal of clustering is to remove attack points that not follow the baseline of major normal data. Attacks should be much lesser than normal traffic, otherwise malicious connection will dominate the normal traffic. We recommend malicious volume below 10% of the total incoming training data. In daily condition, many networks have that upper bound limit. Our experiments use 7000 connections for input data, which are chosen randomly from KDDTrain+ data and include above 900 attack connections.

TABLE I
6-FEATURES USED IN CLUSTERING

Features	Meaning
protocol_type	Protocol types (tcp, udp, …)
src_bytes	Number of bytes from source
count	Number of connections to the same host as the current connection in the past two seconds
diff_srv_rate	% of connections to different services
dst_host_same_srv_rate	% of connections to the same service for destination host
dst_host_serror_rate	% of connections that have SYN errors for destination host

The distance used of K-means algorithm is correlation distance. We use the number of clustering $K=2$ for K-means. The step of clustering input training data are:

1) Choose target cluster for training set.
2) Run K-means several times until target cluster have adequate data point. With 7000 records, the training data should have above 5000 records.

Table II depicted the number of attack connection before and after clustering input data. The result showed that clustering the input data significantly reduced the noise which includes attacks in the background traffic.

TABLE II
REDUCED NOISE OF INPUT DATA

Test	Number of attacks before clustering	Number of attacks after clustering	Attack Reduced Ratio (%)
Test1	928	152	83.60
Test2	937	157	83.20
Test3	935	164	82.50

D. *Experiment with 2-thresholds Method*

After clustering, the targeted cluster becomes the data for training. The system will derive the training parameters needed. For measuring the test accuracy, we use random 50,000 connections in KDDTrain+ data with the same features in table I. Through experiments we found that it is better to keep major PCs $W_1 = 3$ and minor PCs $W_2=3$. Increase or decrease PCs more or less than 3 will make the total accuracy decrease.

The results before clustering input data are shown in table III. Table IV depicts the same tests after clustering.

TABLE III
RESULT BEFORE CLUSTERING INPUT DATA

Test	Precision (%)	Recall (%)	FPR (%)	Total Accuracy (%)
Test1	86.6	22.7	3	62.6
Test2	87	22	2.83	62.4
Test3	83.6	22.2	3.7	62

Recall rate is sensitive with outlier in the data. Result in table III showed that a small of approximately 900 attack records in total 7000 training records still make the recall result very low accuracy. That make the total accuracy only achieve above 60%. Table IV shows the effective of removing noise with K-means in detection result.

TABLE IV
RESULT AFTER CLUSTERING INPUT DATA

Test	Precision (%)	Recall (%)	FPR (%)	Total Accuracy (%)
Test1	92	84.7	6.3	89.5
Test2	92	81.2	6.1	88
Test3	92	82.1	5.6	88.7

E. *Experiment with 1-threshold Method*

In the next experiment, we use 1-threshold method for detection using 13 features in Table V. We choose these features by experience.

We use some small sets of training data which select from pure normal connections, the first and second set have 1000 connections, the third and fourth set has 500 connections. The detection test with random 60000 connections. At first, we use all PCs (*W= 13*) for training phase and detection phase. Then we only use minor PCs ($W_2 = 3, W_1=0$). The detection result in Table VI and Table VII showed that 1-threshold method has good accuracy even with small training dataset. Using only minor PCs for 1-threshold give better recall rate and overall accuracy. Other advantage is the reduction dimension from 13 to 3 PCs.

TABLE V
13-FEATURES USED IN 1-THRESHOLD EXPERIMENT

Features	Meaning
interval	Interval of the connection
protocol_type	Protocol types (tcp, udp, …)
service	Destination service (e.g. telnet, ftp)
flag	Status flag of the connection
source_bytes	Bytes sent from source to destination
destination bytes	Bytes sent from destination to source
count	Number of connections to the same host as the current connection in the past two seconds
srv_count	Number of connections to the same service as the current connection in the past two seconds
serror_rate	% of connections that have Synchronization errors
rerror_rate	% of connections that have Rejection errors
diff_srv_rate	% of connections to different services
dst_host_count	Count of connections having the same destination host
dst_host_srv_count	Count of connections having the same destination host and using the same service

TABLE VI
1-THRESHOLD METHOD USING ALL PCS

Test	Precision (%)	Recall (%)	FPR (%)	Total Accuracy (%)
Test1	95.2	80	3.5	88.7
Test2	94.7	79.6	3.9	88.3
Test3	94	80.9	4.5	88.6
Test4	93.3	81	5.11	88.4

TABLE VII
1-THRESHOLD METHOD USING 3 MINOR PCS

Test	Precision (%)	Recall (%)	FPR (%)	Total Accuracy (%)
Test1	94.2	88	4.6	92
Test2	93.3	86.4	5.32	91
Test3	92.8	87.1	5.8	91
Test4	92.6	88.7	6.1	91.5

Next we test the detection (Table VIII) with training data after remove noise by K-means. The detection result is still good with 1-threshold method. We believe that clustering step remove some normal data with higher distance that make the false positive ratio above 10%.

TABLE VIII
1-THRESHOLD RESULT AFTER CLUSTERING

Test	Precision (%)	Recall (%)	FPR (%)	Total Accuracy (%)
Test1	86.2	84.7	11.6	86.7
Test2	86	83.8	11.7	86.2
Test3	86	81	11.2	85.2

In this case, using PCs with higher variance will take more normal data of higher distance. Table IX depicts the result when using all PCs to compute the distance (*W=13*). With more normal data detected, true negative ratio (TNR) which is the ratio between normal data and total normal data increase above 90% (and decrease FPR= 1−TNR).

TABLE IX
USING ALL PCs TO TAKE MORE NORMAL DATA

Test	Precision (%)	Recall (%)	FPR (%)	Total Accuracy (%)
Test1	91.8	83.4	6.5	88.7
Test2	91.9	83.3	6.4	88.7
Test3	91.2	83.5	7.1	88.5

We don't need to use all PCs for this purpose. Table X depicts the result when using 7 *major PCs* ($W_1=7, W_2=0$). As we expected, major PCs have more variance which detect normal data of high distance better. The advantage here is smaller number of necessary dimensions.

TABLE X
USING 7 MAJOR PCs TO TAKE MORE NORMAL DATA

Test	Precision (%)	Recall (%)	FPR (%)	Total Accuracy (%)
Test1	91.7	83.7	6.6	88.8
Test2	91.3	83.8	7.1	88.6
Test3	90.8	84	7.5	88.5

Because noise filter step by K-means often remove high distance data especially in *minor PCs*, K-means should only use with new incoming training data. Normal data feedback after validation will go directly to normal data pool (Fig. 4) awaiting for principal component analysis next time. With the feedback-regulation mechanism of normal data, more high distance data will add to the training dataset and improve the quality of normal profile. When a new profile is created from better quality normal dataset, detection system can use 1-threshold with small minor PCs. 1-threshold method can reduce the computation overhead and delay when analysis large amount of data.

V. CONCLUSION

In this work, we proposed M-PCA method for network traffic anomaly detection. Our approach concentrated on building normal traffic profile of the anomaly detection model. Through experiments we also showed that some features of NSL-KDD dataset are efficient with the normal profile. We propose a K-means clustering algorithm to reduce noise with input training data. The experiments showed that even with small training dataset (less than 1000 points), our approach has good performance including detection accuracy. We also proposed a new model integrates anomaly detection system with signature-based detection system along with some enhancements of building quality normal profile. In our future plan, we will develop and experiment the proposed model with an open source IDS in real network.

REFERENCES

[1] S. Axelsson, Intrusion Detection Systems: A Survey and Taxonomy, Chalmers University of Technology, Goteborg Sweden, 2000. http://www.cs.chalmers.se/~sax/ pub/

[2] S.S. Rajan, V.K. Cherukur, An Overview of Intrusion Detection Systems, 2010. http://www.idt.mdh.se/

[3] V. Jyothana, V. V. R. Prasad, K. M. Prasad, A Review of Anomaly based Intrusion Detection Systems, *Interl. Journal of Computer Applications*, Vol. 28–No.7, August 2011, pp. 28-34.

[4] A. Jain, B.Verma, J. L. Rana, Anomaly Intrusion Detection Techniques: A Brief Review, *Interl. Journal of Scientific & Engineering Research*, Vol. 5, Iss. 7, July 2014, pp.1372-1383.

[5] S. Myers, J. Musacchio, N. Bao, Intrusion Detection Systems: A Feature and Capability Analysis. Tech.Report UCSC-SOE-10-12. Jack Baskin School of Engineering, 2010.

[6] C. Kacha, K. A. Shevade, Comparison of Different Intrusion Detection and Prevention Systems, *Intl. Journal of Emerging Technology and Advanced Engineering*, Vol.2, Iss.12, Dec.2012, pp.243-245.

[7] M.H.Bhuyan, D.K.Bhattacharyya, J.K.Kalita, Survey on Incremental Approaches for Network Anomaly Detection, *Journal of Communication Nteworks and Information Security (IJCNIS)*, Vol.3, No.3, pp. 226-239, Dec.2011.

[8] M.L. Shyu, S.C.Chen, K.Sarinnapakorn, L.W.Chang, A Novel Anomaly Detection Scheme Based on Principle Component Classifier, *Proc. of the IEEE foundation and New Directions of Data Mining Workshop in 3rd IEEE Intl. Conference on Data Mining (ICDM03),* pp. 172-179, 2003.

[9] D. Brauckhoff, K. Salamatian, and M. May: Applying PCA for Traffic Anomaly Detection: Problems and Solutions, *IEEE INFOCOM*, pp.2866-2870, April, 2009.

[10] H.Huang, H.Al-Azzawi, H.Brani, Network Traffic Anomaly Detection, ArXiv:1402.0856v1, 2014.

[11] C.A.P.Boyce, A.N.Zincir-Heywood, A Comparison of Four Intrusion Detection Systems for Secure E-Business, 2003.

[12] V.J.Hodge, J.Austin, A survey of outlier detection methodologies, 2004.

[13] The KDD Archive, KDD99 cup dataset, 1999, http://kdd.ics.uci.edu /databases/kddcup99/kddcup99.html

[14] NSL-KDD data set for network-based intrusion detection systems, http://nsl.cs.unb.ca/KDD/NSLKDD.html, March 2009.

[15] M.K. Siddiqui, S.Naahid, Analysis of KDD CUP 99 Dataset using Clustering based Data Mining, 2013.

[16] M. Tavallaee, E. Bagheri, W. Lu, A.A. Ghorbani, A Detailed Analysis of the KDD CUP 99 Data Set, *In the Proc. of the IEEE Symposium on Computational Intelligence in Security and Defense Applications (CISDA 2009)*, pp. 1-6, 2009.

Nguyen Ha Duong received the B.E. degree in electronic and telecommunication engineering from the Ha Noi University of Technology, Vietnam, in 2001, and the Msc. degree in electronic and telecommunication engineering from the Ha Noi University of Technology, Vietnam, in 2003. In 2001, he joined the Department of Network and System Engineering, IT Faculty, National University of Civil Engineering as a lecturer. His current research interests include network security, network protocol, routing, data mining and machine learning.

Hoang Dang Hai received the Diplom-Ing. degree in Technical Cybernetics from the Technical University Ilmenau (Germany) in 1984, Dr.-Ing. degree in Telematics and Dr.-Ing.habil. degree from the Technical University Ilmenau (Germany) in 1999 and 2003, respectively. He is currently an Associate Professor at the Post and Telecommunication Institute of Technology (PTIT), Ministry of Information and Communications of Vietnam. His current research interests include information security, wireless sensor networks, network security and network traffic management.

Ontological-semantic text analysis and the question answering system using data from ontology

Kuznetsov V. A.*, Mochalov V. A.**, Mochalova A. V.**

* *Petrozavodsk State University, Lenina str. 33, 185910 Petrozavodsk, Russia*

** *Institute of Cosmophysical Research and Radio Wave Propagation FEB RAS, Mirnaia str. 7, 684034 Paratunka, Kamchatka region, Russia.*

kuznetcv@mail.ru, sensorlife@mail.ru, stark345@gmail.com

***Abstract*—Today, with an avalanching increase in information, the task of developing the systems that allow user to quickly search desired information in large text volumes is becoming more and more urgent. An example of such system is the question answering one. In the work we describe an architecture of such system which work is based on utilizing data from an ontology. We propose an algorithm for automatic update of the ontology basing on use of an expert system and ontological rules for logical inference. We also describe an ontology with the structure based on object-oriented model and describe the functions that are used to update the ontology and extract data from it. We describe the way to update the ontology and to modify the stored data using the rules stored in the ontology. For writing ontological-semantic rules we use the Drools expert system that utilizes the PHREK algorithm for fast pattern matching.**

We analyze the issues of using Apache Spark system for distributed implementation of the algorithm.

***Keyword*—Question Answering System, ontology, expert systems, semantic analysis.**

I. INTRODUCTION

DEVELOPMENT of question answering systems is nowadays becoming more and more urgent problem. This is connected with an avalanche growth of information volume that modern people need to operate.

Basing on analysis of operation algorithms of many Question Answering Systems (QASs), including Lasso [1], QA-LaSIE [2], TEQUESTA [3] etc., one can conclude that all of them do, in the whole, comply a certain general architecture. The high-level representation of the latter is given at Fig. 1. The system receives a question on the natural language as an input. After that, the text of the question is automatically processed. The main stages of this process are preliminary text processing, tokenization, morphological, syntactic and semantic analysis, extraction of named entities and definition of the logical links between the parts of the sentences. Some of these stages may be dropped out or simplified in different QASs implementations and descriptions. Some are, on the contrary, basical for the system operation in a whole: an example is semantic analysis in the work of M. V. Mozgovoy [4]. Several more procedures that can be executed on the stage of automated question text processing are definition of the question type, definition of the expected answer type etc. Basing on results of the automated question text processing, a query is formed to be passed on to a search engine. The search engine, in its turn, selects a predefined number (N) of documents most relevant to a query from the collection. The texts of each of selected documents as well as the question text are automatically processed. Here, the machine algorithms of the question text processing may differ from the algorithms of the selected documents processing. Further, by means of internal algorithms of the QAS, the specific text fragments are selected from the documents returned by the search engine. The selected text fragments are presented by the system as an answer. The most advanced QASs can use data from factbases (FB), databases (DB) and ontologies on the stage of text fragments selection. The information from such data storages can complement the answer/answers of the system.

In the work we propose an operation algorithm of an ontological-semantic analyzer (a semantic analyzer that uses ontology) of text. We describe how to use results of its work in a developed question answering system. During operation of the ontological-semantic analyzer, the ontology is being changed and its data is used to define semantic links between parts of the sentences. As a result of these changes, the data in the ontology may be learned, deleted or updated. In such a way the ontological-semantic analysis solves one more task apart from being used in a question answering system: it performs automated ontology learning.

Further we give a brief overview of the methods of ontologies learning.

In work [5] the authors define three main methods of

Manuscript received July 9, 2015. The work was implemented with financial support from the Russian Foundation for the Humanities as part of research project No.15-04-12029 “Software development of an electronic resource with an online version of a Russian-language question answering system”.

Kuznetsov V. A. is with the Petrozavodsk State University, Russia (e-mail: kuznetcv@mail.ru) .

Mochalov V. A., is with the Institute of Cosmophysical Research and Radio Wave Propagation FEB RAS, Russia (e-mail: sensorlife@mail.ru).

Mochalova A. V. is with the Institute of Cosmophysical Research and Radio Wave Propagation FEB RAS, Russia (corresponding author to provide phone: 8-916-820-15-25; fax: 8-(41531)-33718; e-mail: stark345@gmail.com).

ontologies learning:

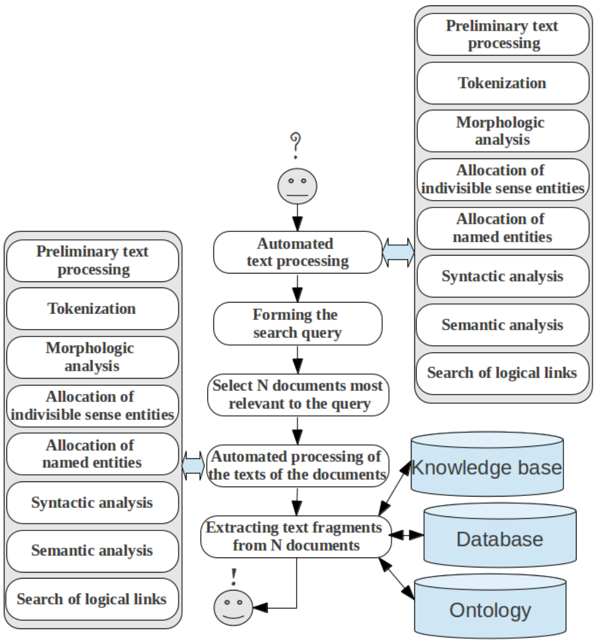


Fig. 1. Workflow architecture in a question answering system

1) «manual» input;
2) automatic or automated input using traditional lexicographical information (encyclopedic, word-defining and other dictionaries and databases);
3) automatic or automated input based on analysis of distributive characteristics of the lexis in text corpus.

Ontology learning by means of manual input is a very labor consuming procedure that requires participation of highly qualified specialists. For this reason, development of automatic (or at least partly automated) methods of ontology learning is today a very urgent task.

One of the most frequent methods of automatic ontology learning is constructed on analysis of dictionary definitions. In such methods, the ontological constructions (entities and the type of their relation) discovered by use of pattern search in dictionary definitions are added to the ontology. For instance, if one needs to collect information about all possible means of transport, the dictionary search is performed using all patterns of the kind “X is a vehicle that...”, “X is a means of transport used for...”, “X is a vehicle type equipped for...” etc. Such search patterns may be developed manually or automatically using self-learning programs [6]. The idea of such a way of automated ontology learning is described in works [7], [8], [9], [10].

The authors of the work [11] compare the methods of automated ontologies learning, consider their own experience and confirm that at the present day the most promising technology from the perspective of obtaining practical results is the one that uses traditional lexicographical information (encyclopedic and word-defining dictionaries).

The work [12] describes a method of automatic construction of domain-specific ontology basing on analysis of linguistical characteristics of text corpus.

A more detailed survey of the existing methods of automated ontologies learning is provided in work [13].

II. Ontology Structure and Functions for Working with the Ontology

An **ontology** (in a formulation by Gruber who was one of the first to use this notion in the area of information technology) is a formal specification of conceptualization [14]. By conceptualization, we understand a description of a set of notions (concepts) of the subject domain and links (relations) between them. As understood today, ontology in informatics is an hierarchical data structure including all relevant object classes, their connections and rules (restrictions) defined in this domain and necessary for solving the problems assigned for an information system.

In this work we propose to use an object-oriented ontology model consisting of **classes** (certain abstract images sharing in the common sense a certain set of properties) and **objects** (part of the real world having some certain properties of the class it belongs to; the same object may belong to different classes and the same class may have different objects). Classes and objects can be interconnected by various **relations** that define certain dependencies between them. Here, a sequence of arguments, for which the relation is being defined, does matter. Examples of such structured data storages are DBpedia [15], Freebase [16] and Wikidata [17], Wikipedia [18], Wiktionary [19]. For instance, in work [20] the authors propose an approach to the automated construction of a general-purpose lexical ontology based on the Wiktionary data. When adding new information (classes, objects or relations between them) to the ontology, one needs to store the timestamp of the changes being made and the source basing on which the changes are made. It should also be possible to make a recovery of the ontology modifications (for example, when an error has been detected). To make this possible, the whole history of ontology modification should be stored. The initial ontology structure can be formed using ideographic dictionaries.

To be specific, we describe high-level functions for ontology learning with the data and for data extraction from the ontology:

1) **CreateClass (idClass, nameClass)** – adds a class with unique identifier idClass and the name nameClass into the ontology;
2) **CreateObject (idObject, idClass, nameObject)** – adds an object with unique identifier idObject and the name nameObject of the class with unique identifier idClass into the ontology;
3) **CreateRelation (idRelation, relationName)** – creates a relation with unique identifier idRelation and the name relationName;
4) **CreateRelation (id1, id_2, idRelation, h)** – adds a relation with unique identifier idRelation, linking two classes or objects (or an object and a class) with unique identifiers id_1 and id_2 correspondingly, where h is a frequency (or weight) of the relation, into the ontology;
5) **Inheritance (id1, id2)** – adds information that the class with unique identifier id2 is inherited from the class with unique identifier id1 into the ontology;
6) **Inheritance_with_denial (id1, id2, {[idRelation1], [idRelation2], [idRelation3], …})** – adds information that the class with unique identifier id2 is inherited from the class with unique identifier id1 excepting relations with unique identifiers from the array {[idRelation1], [idRelation2], [idRelation3], …} into the ontology;

7) **PartialInheritance (id1, id2, {[idRelation1], [idRelation2], [idRelation3], …})** – adds information that the class with unique identifier id2 is inherited from the class with unique identifier id1, but only the relations with unique identifiers from the array {[idRelation1], [idRelation2], [idRelation3], …} are inherited, into the ontology;
8) **ReturnClassName (idClass)** – returns the name of the class with unique identifier idClass;
9) **ReturnObjectName (idObject)** – returns the name of the object with unique identifier idObject;
10) **What_Is (text, position)** – returns value 1 if the sense entity from the analyzed text text, having its position in this text is explicitly defined by the data stored in position variable, is an ontology class; 2 if this entity is an ontology object; 3 if there has been detected no information about this sense entity in the ontology;
11) **ReturnAllRelations (idClassOrIdObject)** — for a class or an object with unique identifier idClassOrIdObject, returns a set consisting of the following values groups:
    a) idRelation – unique identifier of the relation linking a certain class or an object to the class with unique identifier idClass;
    b) idClassRel – unique identifier of a class or an object to which the class with unique identifier idClass is linked by the relation with unique identifier idRelation;
    c) relationName – relation name;
    d) val – relation value;
    e) h — relation frequency (or weight);
12) **ReturnRelations (idClassOrObject, RelationName)** – is similar to the function ReturnAllRelations (idClassOrIdObject) in which the names of relations are restricted to RelationName only;
13) **ReturnAllObjects (idClass)** — returns a set of unique identifiers of objects from the class with unique identifier idClass;
14) **ReturnParentClasses (idClassOrObject)** — returns unique identifiers of the classes that are parent to a class or an object with unique identifier idClassOrObject;
15) **ReturnRelationValue (idClassOrObject1, idClassOrObject2, relationId)** – returns the value of the relation with unique identifier relationId, linking two objects or ontology classes with unique identifiers idClassOrObject1 and idClassOrObject2;
16) **ReturnAllAncestorClasses (idClassOrObject)** – returns unique identifiers of all classes that are ancestors to a class or an object with unique identifier idClassOrObject;
17) **ReturnAllSuccessorClasses (idClass)** – returns unique identifiers of all classes that are successors of any level for a class with unique identifier idClass;
18) **ReturnAllSuccessorObjects (idClass)** – returns unique identifiers of all objects of the classes that are successors of any level for the class with unique identifier idClass and for this class itself.
19) **insertOrUpdateRelation(RelationName, idClassOrObject1, idClassOrObject 2, h)** – if the ontology does already contain the relation named RelationName, connecting the class or the ontology object with unique identifier idClassOrObject1 to the class or the ontology object with unique identifier idClassOrObject2, then increase the weight of this relation by h. If the ontology does not contain such relation, then add it and assign the weight h.

For writing ontological-semantic rules we propose to use the following notations:
1) **Unit** – undefined token, class, object or relation from the ontology;
2) **UnitCO** – class or object from the ontology;
3) **Class** – class from the ontology;
4) **Object** – object from the ontology:
5) **Rel** – relation linking two classes, two objects or a class and an object from the ontology;
6) **UDT** – undefined token, that is a token (indivisible sense entity) the information about which is not present in the ontology yet. That means that this token is neither an ontology class nor an object.

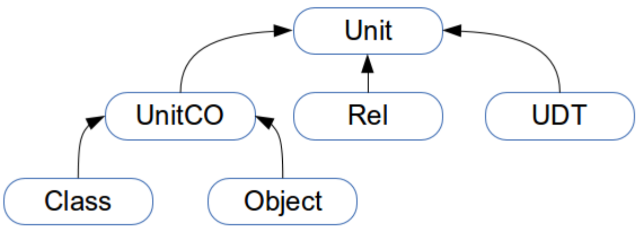


Fig. 2. Hierarchy of the notations used for describing ontological-semantic rules.

Hierarchical relationship between these notations is given at Fig. 2.

III. Architecture of the Question Answering System Integrated with an Ontology

The question answering system is supposed to operate in two modes: ontology learning mode (when system input are valid texts and the general ontology is being modified) and user answering mode. By "valid" here we understand the text containing the data the verity of which is unquestioned. In the scheme presented at Fig. 3 and depicting an architecture of the question answering system, everything concerning only the user answering mode is marked with a dash line.

Ontology learning mode

When a question answering system operates in ontology learning mode, it receives a valid text T as its input. The text enters the module of initial text processing where it is preliminarily processed: text formatting symbols that do not carry any role during text analysis are removed, orthographical and syntactic mistakes are corrected, extra symbols of whitespaces and line breaks are removed and so on. The tokenization stage follows, including breaking text into paragraphs, sentences and words. For each marked word its morphological properties are being defined with use of corresponding morphological dictionaries. The next stage is separating non-divisible sense entities which can be separate words or groups of words that are united by some common meaning. Some examples of named entities consisting of several words are some named entities (New York, Santa Claus, Mr. Smith etc.) or composite parts of speech (an idiom "good and proper", a linking word "such as", a numeral adjective "forty five" etc.). The final stage of the initial processing is search for logical links in the text.

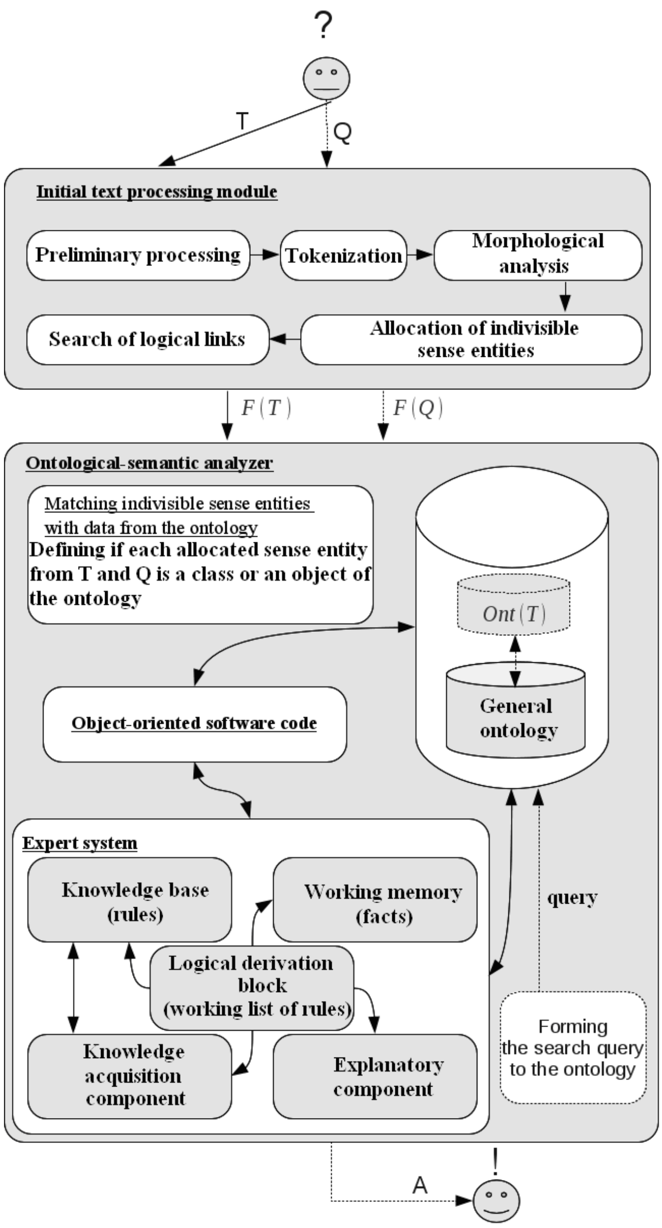


Fig. 3. Architecture of the question answering system.

The text T that has been initially processed, together with all data received on this stage, is denoted as F(T) on the scheme and passed onto input of ontological-semantic analyzer.

The work of ontological-semantic analyzer begins with association of non-divisible sense entities separated in the valid text T with classes and objects from the ontology. At this stage, a task of lexical ambiguity resolution is being solved for the purpose of defining which one of the set of existing classes and/or objects with the same names does the considered sense entity belong to.

After the valid text has been initially processed, the ontology modification takes place. For this, the ontology itself, the expert system (consisting of the Knowledge base containing rules, the Working memory containing facts, the Block of logical derivation containing the operational list of the rules and the Component of knowledge acquisition) and the object-oriented code are used. As a result of certain rules stored in the expert system (ES), the ontology may experience addition or removal of certain classes or objects, as well as addition, removal or modification of the relations between them.

User answering mode

During operation of the question answering system, the system receives as its input the question Q asked by a user and the text T selected by the user for searching the answer in. The user provides the question sentence Q in natural language. The text T to search the answer in and the question Q are passed onto input of the module of initial text processing where they get through the same stages of automated text processing that were described for the ontology learning mode. The working results of this module (F(Q) for the question sentence and F(T) for the text in which an answer is searched), on the analogy with the ontology learning mode, are passed onto input of the module of the ontological-semantic analyzer.

In the ontological-semantic analyzer, the association of non-divisible sense entities that have been separated in Q and T with classes and objects of the ontology takes place.

Next, with use of the Ontology and the Expert system, the separate ontology Ont(T) is created by means of the object-oriented code after the text that has been provided by the user and experienced the initial processing. All classes and objects of the ontology Ont(T) refer to the classes and objects from the main ontology but do not modify the latter. No modification of the common ontology is performed when the system operates in user answering mode.

The next stage of work of the ontological-semantic analyzer in the considered mode is formation of a search query to the ontology basing on existing information about Q and T. The search query is formed using functions describing work with the ontology (see p.1.) and the Unit-terminology, the hierarchical dependency of which is presented in Fig. 2.

When developing the query to the ontology, one should consider not only the user-defined question Q itself, but also the text T in which an answer is to be found. It is related to the fact that the user-provided text can help in solving the tasks of lexical ambiguity that can arise when forming the query. For example, if the question contains the word “bank” which may mean either “coast” or “credit institution”, then the user-provided text T may help to solve the lexical ambiguity that arose in the question. It is more likely that the word “bank” is used in the meaning “coast” if the text T contains the words which are such classes or objects in the ontology that are close to the class or the object “bank” in the meaning “coast” of a river, a sea etc.

In case the query to Ont(T) has been successfully executed and returned data from the ontology, these data are provided to the user as an answer. In case an answer has not been found in Ont(T) or it has not satisfied the user, the search is continued in the general ontology. Using the explanatory component of the expert system, the user can learn how the system has obtained the solution.

IV. Ontology Learning Mode

This section will describe the working algorithm of the ontological-semantic analyzer operating as a part of the question answering system which is functioning in the ontology learning mode.

A semantic dependency is a certain universal relation that a native speaker beholds in the language. This relation is binary, that is, it holds from one semantic node to another [21]. It is convenient to regard indivisible sense entities of the language as semantic nodes. They can be represented, for example, by the named entitites. We say that two different semantic nodes α and β from the same sentence are related by a semantic dependency named R (denote $R(\alpha,\beta)$) if there is a certain universal binary relation between α and β.

For concrete semantic nodes α and β and the dependency R, the direction is selected in such a way that the formula $R(\alpha,\beta)$ would be equivalent to the statement that "β is R for α".

Queue with priority

In a classical definition a queue with priority is defined as an abstract data type allowing to store pairs (key, value) and supporting the following operations [22]:

1) **init** — initialize a new empty queue;
2) **insertToPriorityQueue** — insert a new element into the queue;
3) **remove** — remove and return the highest-priority element of the queue.

In this work we will use the "queue with priority" for removing facts from the working memory of the expert system. The fact of the expert system consists of the following: UnitCO, a link to the previous fact (left) and to the next fact (right), morphological characteristics and coordinates in text (the number of the sequence and the position of UnitCO in the sentence). In this context we use two values to define the element priority in the queue:

1) priority of the group that the considered semantic relation belongs to;
2) position of ***UnitCO*** in the analyzed sentence.

In this work we will consider the queue **Q** with priority, the elements of which will consist of the following triples:

1) ***a*** – the name of UnitCO;
2) ***sp*** – the priority of a semantic group which the semantic relation belongs to, one of the arguments being UnitCO;
3) ***pos*** – the position of UnitCO in the analyzed sentence.

We will say that the element ***(a, sp, pos)*** of the described queue ***Q*** has the highest priority if ***sp*** value of this element is minimal and ***pos*** value is maximal. Consequently, the elements of the queue with priority are sorted in ascending order of priorities of semantic dependencies groups. If the queue contains several elements with the same priority values of semantic groups, then such elements are sorted in descending order by the last ***UnitCO*** related to the considered element in the analyzed sentence.

Below we describe the rules to add the element ***(a', sp', pos')*** into the queue Q with priority for the case when Q contains the element ***(a, sp, pos)*** such that ***(a = a')*** and ***(pos = pos')***:

1) ***(sp'> sp)***, hence ***Q = Q\(a, sp, pos)U(a', sp', pos')***;
2) ***(sp'<= sp)***, hence ***Q*** is not modified.

Basical ontological-semantic patterns

Let us call an ontological-semantic pattern the rule by which the expert system finds semantic dependencies between classes and objects in the analyzed text (where indivisible sense entities are marked and each of them is refered to a certain class or an object of the ontology). A basical ontological-semantic pattern which is a rule of the expert system consists of the left and the right sides. The left side of the pattern describes the conditions upon which the actions described in the right side are executed. So, for example, the left side of the pattern always describes a biconnected facts list, and can also describe boolean functions having the facts from that biconnected list as their arguments.

The right side of the pattern contains the list of actions each of which can: modify any fact of the ES (by modifying the relation in a corresponding UnitCO); queue for removal a fact of the ES having a certain priority of removal; other actions.

Below we provide an example of a basical ontological-semantic pattern that describes how the ontology is modified if in the analyzed text there has been discovered a fact X containing UnitCO being a class of the ontology (defined as a Property in the ontology and containing among the morphological characteristics the information that it is the singlular number), and what follows next is some fact Y containing UnitCO being a class or an object of the ontology, and it is known that in the text it is preceded by the fact X, and among the morphological characteristics of Y there is information that it is the singlular number of a noun.

```
when
{
$X : Fact (unitCO.type == "Class",
unitCO.hsOntAttrs contains "Property",
hsMorphAttrs contains "singular")
$Y : Fact (unitCO.type == "Object" ||
unitCO.type == "Class" , prev == $X,
hsMorphAttrs contains "singular", hsMorphAttrs
contains "noun")
}
then
{
Logic.insertOrUpdateRelation("Property",
$Y.unitCO.id, $X.unitCO.id, 1);
Logic.insertOrUpdateRelation("Property for",
$X.unitCO.id, $Y.unitCO.id, 1);
Logic.insertToPriorityQueue ($X, sp);
}
```

After all basical ontological-semantic patterns (with true left sides) have been found on the current facts of the expert system, one fact with the highest priority is removed from the queue for removal and from the working memory of the expert system. When removing a fact from the working memory of the expert system, one should update the left and the right facts for the fact being removed (as shown in Fig. 4).

The Table 1 shows an example of using the ontological-semantic analyzer and how the priority queue (Q) is gradually changing. The analyzed text (AT) is "Yesterday, the yachting sport school honors left for a camp". Algorithm of ontology learning using the basical ontological-semantic patterns

Used notations:

1) P – the analyzed sentence;
2) p_i – i-th indivisible sense entity of the analyzed sentence P;
3) S – the set of all rules of the ES including the basical ontological-semantic patterns;
4) S_i – i-th rule of S;
5) Q – the queue with priorities.

TABLE I
AN EXAMPLE OF USING THE ONTOLOGICAL-SEMANTIC ANALYZER

Step	Found semantic relationships	Operations on Q	Elements of Q
1	AT: [0] [Yesterday], [1] [the yachting] [2] [sport] [3] [school] [4] [honors] [5] [left] [6] [for] [7] [a camp].		
	Belong(honors, school)	insert(Q,(school, 2, [3]))	Q = {(school, 2, [3])}
	Belong(school, sport)	insert(Q,(sport, 2, [2]))	Q = {(school, 2, [3]), (sport, 2, [2])}
	Property(sport, the yachting)	insert(Q,(the yachting, 1, [1]))	Q = {(school, 2, [3]), (sport, 2, [2]), (the yachting, 1, [1])}
	Location(left, for a camp)	insert(Q,(for a camp, 3, [7]))	Q = {(school, 2, [3]), (sport, 2, [2]), (the yachting, 1, [1]), (for a camp, 3, [7])}
	Action(left, honors)	insert(Q,(honors, 15, [4]))	Q = {(school, 2, [3]), (sport, 2, [2]), (the yachting, 1, [1]), (for a camp, 3, [7]), (honors, 15, [4])}
		remove(Q, (the yachting,1,1))	Q = {(school, 2, [3]), (sport, 2, [2]), (for a camp, 3, [7]), (honors, 15, [4])}
2	AT: [0] [Yesterday], [2] [sport] [3] [school] [4] [honors] [5] [left] [6] [for] [7] [a camp].		
	(new semantic relationship is not found) AND isEmpty (Q) = false	} =>	Continue
		remove(Q, (school, 2, [3]))	Q = {(sport, 2, [2]), (for a camp, 3, [7]), (honors, 15, [4])}
3	AT: [0] [Yesterday], [2] [sport] [4] [honors] [5] [left] [6] [for] [7] [a camp].		
	(new semantic relationship is not found) AND isEmpty (Q) = false	} =>	Continue
		remove(Q, (sport, 2, [2])	Q = {(a camp, 3, [7]), (honors, 15, [4])}
4	AT: [0] [Yesterday], [4] [honors] [5] [left] [6] [for] [7] [a camp].		
	(new semantic relationship is not found) AND isEmpty (Q) = false	} =>	Continue
		remove(Q, (a camp, 3, [7]))	Q = {(honors, 15, [4])}
5	AT: [0] [Yesterday], [4] [honors] [5] [left] .		
	(new semantic relationship is not found) AND isEmpty (Q) = false	} =>	Continue
		remove(Q, (honors, 15, [4]))	Q={}
6	AT: [0] [Yesterday], [5] [left] .		
	Time (left, yesterday)	insert(Q, (yesterday, 7, [0]))	Q={(yesterday, 7, [0])}
		remove(Q, (yesterday, 7, [0]))	Q={}
7	AT: [5] [left] .		
	(new semantic relationship is not found) AND isEmpty (Q) = true	} =>	End

Aiming to check the usage possibilities of Apache Spark platform, we have implemented the programs for the latter. The first program computed in a distributed manner the inverted text index using the Lucene library on a 10-nodes cluster. After that, a second program executed in a distributed manner a large number of search queries using Lucene language to the distributed inverted index. The working results of the programs allowed to establish the following: a) a significant profit in working time of distributed operations of indexing and search in comparison with performing the same operations on a single computer; b) reasonability to implement the proposed algorithm on the Apache Spark platform.

Algorithm:

Step 1. Put in the input of the ontological-semantic analyzer the preprocessed sentence P (see the architecture of the question answering system) in natural language and consisting of indivisible sense entities p_i: $P = (p_1, p_2, p_3, \ldots, p_N)$.

Step 2. Match the indivisible sense entities $p_1, p_2, p_3, \ldots, p_N$ with the ontology data: associate with each p_i some $UnitCO_i$ — a class or an object from the ontology. Form the facts of the ES P_{Facts}, presented using a double-linked list where the first and the last facts ($Fact_0$ and $Fact_{N+1}$) are empty, and the rest are formed according to the fact definition: that is, $Fact_i$ contains $UnitCO_i$, the morphological information about $UnitCO_i$, the links to the left and the right facts, the fact position in the text.

Step 3. Add to the knowledge base of the expert system (S) all ontological-semantic patterns and other rules.

Step 4. Initialize the variable i which will store the sequential number of the considered rule as zero: i := 0.

Step 5. Using fast patterns matching algorithm, form the array G consisting of pairs (rule, fact list) in which we will put the rules from S and the corresponding facts from P_{Facts} with true left part.

Step 6. Sort the elements of the array G in the order of fulfilling the rules by the block of logical derivation.

Step 7. Check whether the value of the loop variable i has exceeded the limits of array G (i < |G|), which would mean that all elements of G has been looked through. If i < |G|, go to Step 8. Else go to Step 10.

Step 8. According to the right side of the rule G_i the following actions may be performed: update the facts of the ES; add facts to the queue for removal together with their priorities; etc.

Step 9. Increase the value of the loop variable i by 1: i := i + 1. Go to Step 7.

Step 10. Check whether the queue for removal Q is empty: if the queue is empty (isEmpty(Q) is true), then finish the algorithm execution.

Step 11. Set the variable b to be equal to the fact being removed from Q and having the highest priority, that is: element b := remove(Q).

Step 12. Update the links of the fact a (a=b.left) and the fact c (c=b.right) in the following way: a.right=c; c.left=a. In the working memory of the expert system: update the facts a and c; remove the fact b. Go to Step 4.

The proposed working algorithm of the ontological-semantic analyzer can be implemented in distributed manner. For instance, the analyzed texts can be stored in a distributed file system (such as HDFS). The following tasks can be distributed among the nodes of a computational cluster: the ontology construction and learning for each analyzed text; indexing of ontological-semantic graphs; search for the answers to the questions; etc. A promising platform for distributed implementation of the proposed algorithm on a cluster may be the system Apache Spark. This platform has already been integrated in the Hadoop ecosystem (HDFS, Hadoop YARN) and is a part of such popular integration projects as Cloudera, HortonWorks, MapR etc.

Aiming to check the usage possibilities of Apache Spark platform, we have implemented the programs for the latter. The first program computed in a distributed manner the inverted text index using the Lucene library on a 10-nodes cluster. After that, a second program executed in a distributed manner a large number of search queries using Lucene language to the distributed inverted index. The working results of the programs allowed to establish the following: a) a significant profit in working time of distributed operations of indexing and search in comparison with performing the same operations on a single computer; b) reasonability to implement the proposed algorithm on the Apache Spark platform.

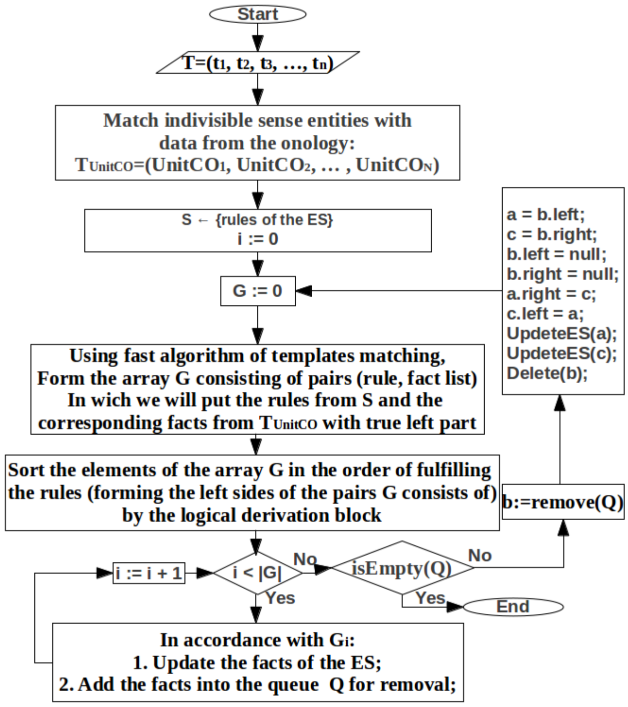


Fig. 4. Logical diagram of the algorithm of automated ontology learning.

V. Conclusion

The work is devoted to development of architecture and prototype of a question answering system that uses data from the ontology. The structure and the functions to work with the ontology are described in this paper. A working algorithm of an ontological-semantic analyzer using basical ontological-semantic patterns with removal is programmatically realized in Java language. The program has been registered in the Rospatent [23]. The Drools system, that uses a fast pattern matching algorithm with PHREAK patterns, has been used as the expert system. Using the expert system Drools has ensured high working speed of the ontological-semantic analyzer. For instance, the described algorithm of the ontological-semantic analysis using the Drolls expert system and 2160 basical ontological-semantic patterns has determined 8213 semantic relations in 6390 ms in the text of E. T. A. Hoffmann's fairy tale "The Golden Pot". Without the Drolls expert system, the implementation of the algorithm of the ontological-semantic analyzer works in average 6-8 times slower. The experiments were performed on an Intel Core i3 M CPU 2.27 GHz under OS Ubuntu 12.04.

References

[1] Moldovan, D. I., Harabagiu, S. M., Pasca, M., Mihalcea, R., Goodrum, R., Girju, R., and Rus, V.: LASSO – A Tool for Surfing the Answer Net. *TREC-8*, pp. 175-183, 1999.

[2] Scott, S., Gaizauskas, R.: QA-LaSIE: A Natural Language Question Answering System. *Proceedings of the 14th Biennial Conference of the Canadian Society on Computational Studies of Intelligence*, 2001.

[3] Monz, C., de Rijke, M.: Tequesta: The University of Amsterdam's Textual Question Answering System. *TREC*, 2001.

[4] Mozgovoy, M. V.: Simple question-answering system, based on Russian semantic analyzer. *Vestnik of the SPb University*, series 10, issue 1, 116-122, 2006.

[5] Rubashkin, V. Sh., Kapustin, V. A.: Using term definitions in encyclopedic dictionaries for automated ontologies learning. *XI All-Russian united conference «The Internet and the modern society»*, SPb, 2008, http://old.conf.infosoc.ru/2008/pdf_CL/Rubashkin&Kapustin.pdf

[6] Mitrofanova, O. A., Konstantinova, N. S.: Ontologies as systems of data storage. *All-Russian competitive selection of the review analytical papers on the priority direction "Information telecommunication systems"*, 54 p., 2008.

[7] Aramaki, E., Imai, T., Kashiwagi, M., Kajino, M., Miyo, K. and Ohe, K.: Toward medical ontology using Natural Language Processing. http://www.m.u-tokyo.ac.jp/medinfo/ont/paper/2005-aramaki-1.pdf

[8] Hovy, E., Knight, K., Junk, M.: Large Resources. Ontologies (SENSUS) and Lexicons. www.isi.edu/natural-language/projects/ONTOLOGIES.html

[9] Rubashkin, V. Sh.: *Ontological semantics. Knowledge. Ontologies. Ontologically-oriented methods of informational text analysis*. Moscow, Fizmatlit, 2013.

[10] Rabchevsky, E. A.: Automatic ontology construction based on lexical-syntactic patterns for information retrieval. *Proceedings of the 11th All-Russian scientific conference RCDL'2009*, Petrozavodsk, 69-77, 2009.

[11] Rubashkin, V. Sh., Bocharov, V. V., Pivovarova, L. M., Chuprin, B. Yu.: The approach to ontology learning from machine-readable dictionaries. *Dialog — 2010*. http://www.dialog-21.ru/digests/dialog2010/materials/html/63.htm

[12] Orobinska, E. A.: Automatic Method Of Domain Ontology Construction based on Characteristics of Corpora POS-Analysis. *Proceedings of the XV All-Russian united conference «The Internet and the modern society» (IMS-2012)*, SPb, 209-212, 2012.

[13] Buitelaar, P., Cimiano, P.: Ontology learning and population bridging the gap between text and knowledge. *Series: Frontiers in artificial intelligence and applications*, V. 167, Amsterdam; Washington, DC : IOS Press, 2008.

[14] Gruber, T. R. A Translation Approach to Portable Ontology specification. Knowledge Acquition, 5(2), 1993.

[15] DBpedia, http://wiki.dbpedia.org

[16] Freebase, http://www.freebase.com

[17] Wikidata, https://www.wikidata.org

[18] Wikipedia, the free encyclopedia, https://ru.wikipedia.org

[19] Wiktionary, the free dictionary, https://ru.wiktionary.org

[20] Krizhanovsky, A. A., Smirnov, A. V.: An approach to the automated construction of a general-purpose lexical ontology basing on the wictionary data [in Russian]. *Izvestiya RAS, Conntrol theory and systems*, No. 2, 53-63, 2013.

[21] Sokirko, A. V.: Semantical dictionaries in automated text processing: as exemplified by the DIALING system. PhD. tech. sci. diss. [in Russian]. Moscow, 2001.

[22] Downey, A. B. *Think Python*. O'Reilly Media, 300 p., 2012.

[23] Mochalova, A. V. *Certificate on the state registration of the computer program "A program of text semantic analysis using basical semantic patterns with removal"*, No. 2015613430, Russia. 28.01.2015

Vladimir Kuznetsov is a scientist, mathematician, Professor, Doctor of technical sciences, Honorary worker of higher professional education of the Russian Federation, Honored worker of science of the Republic of Karelia. He is the author of over 200 scientific papers in the field of mathematical modeling and solution of applied optimization problems, as well as academic literature on the programming. He is one of the coaches of the teams who were winners of the world championship on programming (in 2007, 2008, 2010). His research interests include optimization techniques, Olympiad programming.

Vladimir Mochalov was born in Lyubertsy, Russia in 1985. He received the Ph.D. degree in electronic engineering from Moscow Technical University of Communications and Informatics. His research interests include networks structure synthesis, artificial intelligence, bio-inspired algorithms, query answering systems and Big Data.

Anastasia Mochalova was born in Petrozavodsk, Russia, in 1987. She received the bachelor's degree at Petrozavodsk State University, the master's degree in St. Petersburg State University of Aerospace Instrumentation. She is an external PhD student in technical sciences at Petrozavodsk State University. Her research interests include automated processing of natural language texts, development of question-answering systems, automation of ontologies creation, development of the semantic analyzer.

Functional scheme of the flying sensor networks architecture design

Mochalov V.A.*, Pschenichnikov A.P.**

*Institute of Cosmophysical Research and Radio Wave Propagation FEB RAS , Mirnaia str. 7, 684034 Paratunka, Kamchatka region, Russia

**Moscow technical university of communications and informatics, Aviamotornaya str. 8a, 111024 Moscow, Russia

sensorlife@mail.ru, pshenichnikov@mtuci.ru

***Abstract*— The process of flying sensor networks (FSN) construction is quite complicated. To date, there has been developed a number of methods and algorithms of solving separate problems arising in the process of FSN construction, but the process itself is not formalized as a rigid set of rules, algorithms and standards following which would guarantee construction of a FSN satisfying the designer's requirements. Many problems arising on the mentioned stages of FSN design are NP-complete and cannot be formalized and solved by traditional analytical methods due to fuzziness of task formulation, initial data, criteria and restrictions. In most cases one does not need to derive the optimal solution of the design task; usually the task is reduced to obtaining a spectrum of solutions satisfying design criteria and to selection of an optimal one among them, with interactive participation of the designer in the process of decision making. In this work we propose the functional scheme of the FSN architecture design, which can be the basis for a specialized design support system of flying sensor networks.**

***Keyword*— flying sensor networks, architecture construction, design support systems.**

I. INTRODUCTION

FLYING sensor networks (FSNs) is used for monitoring and controlling the arduous zone and the rural area in the last time. The applications can include video dissemination via FSN, military cases. The comparison of different protocols, methods for reducing energy consumption, and data synchronization method are the more investigation areas for FSN [1].

Flying sensor networks include at least two network segments: the ground one and the flying one [2] (for example Fig. 1). The ground segment consists of a set of independent geographically-distributed wireless sensor networks (WSNs). Information from the latter is collected by mobile robots (in most applications by unpiloted flying devices and sometimes above-ground, above-water and underwater devices). Apart the functions of information collection, mobile robots may perform: work management of the ground WSN; point allocation; dissemination over the territory; movement, removal and reprogramming of the WSN nodes; charge and replace of power sources etc.

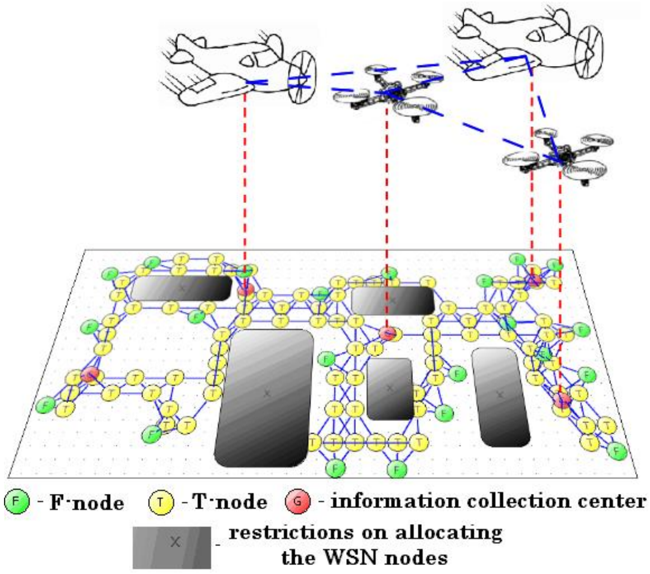


Fig. 1. Example of the flying sensor network

Development of network nodes	Structure and network topology	Connectivity
Network protocols	Routing and addressing	Reliability, survivability and safety
Protection against unauthorized access	Power consumption. Increasing the lifetime of network	Fault tolerance
Formalization of the objectives tree and optimization parameters	Data aggregation	Increasing the accuracy of the collected information
Coverage	Self-configuration, Self-healing, Self-optimization	Automated decision support
Polling	Flow control	Clustering
Interaction with mobile robots	Simulation modeling	QoS
Exact and approximate evaluation of optimization parameters	Three-dimensional space	Moving and removing nodes
Reprogramming nodes	Charging nodes	Positioning

Fig. 2. Several problems arising when constructing the flying sensor network ground structure.

Manuscript received July 9, 2015. This work was supported by RFBR, research project No15 07-09431A "Development of the principles of construction and methods of self-organization for Flying Ubiquitous Sensor Networks".

Mochalov V. A. is with the Institute of Cosmophysical Research and Radio Wave Propagation FEB RAS, Russia (corresponding author to provide phone: 8-925-833-71-98; fax: 8-(41531)-33718; e-mail: sensorlife@mail.ru).

Pschenichnikov A.P. is with the Moscow technical university of communications and informatics, Russia (e-mail: pshenichnikov@mtuci.ru).

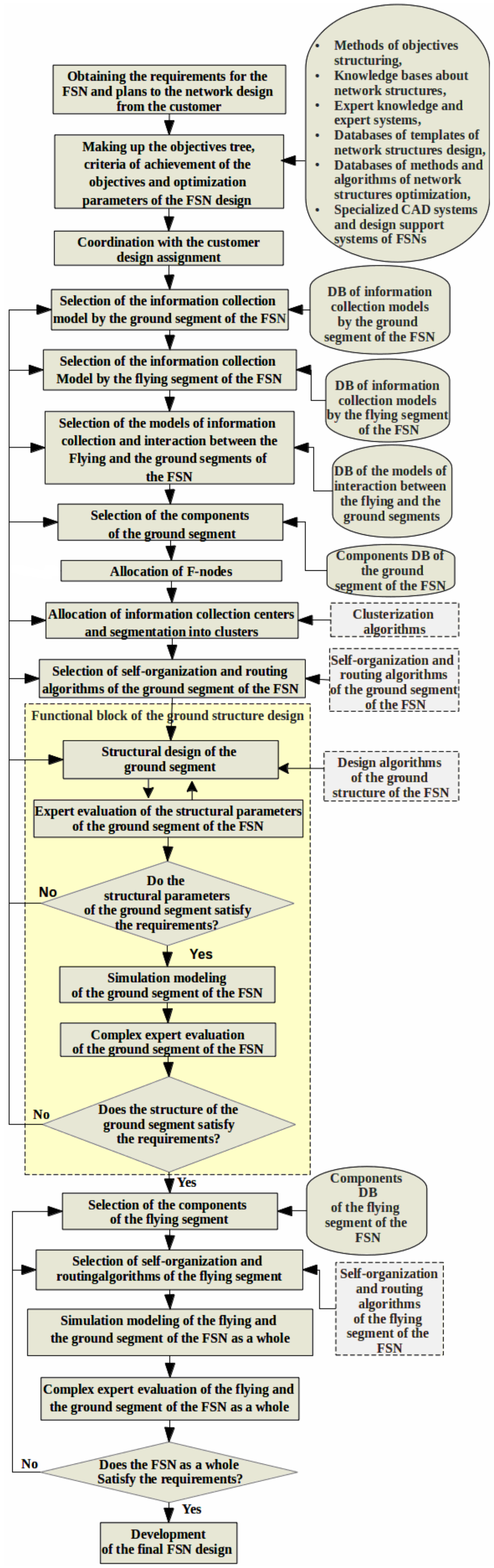


Fig. 3. Functional scheme of the flying sensor networks architecture construction

Construction of the ground segment of FSNs requires solving various complicated problems that relate to different research areas. Some of such problems are given in Fig. 2.

In the work we use a model of the of the ground segment of FSN structure, where on the functional level the following types of nodes can be defined: (1) functional nodes (F-nodes) that collect information in some neighborhood of their location; (2) transit nodes (T-nodes) that manage routing and retransmit the information collected by F-nodes to the information collection centers (ICC) to be utilized further; (3) ICCs that manage the WSN and process information collected by the WSN. In general case there can be multiple ICCs in the WSN, and the information that has arrived into each of them is available to one or multiple users for making decisions and performing certain actions. It means that information received by F-nodes should be retransmitted, with a required degree of reliability, to several ICCs by means of transit nodes allocated within the given object in a certain way.

II. Functional Scheme of the Flying Sensor Networks Architecture Construction

A promising approach to FSN design is use of interactive decision support systems (DSS) [3]. The notion of the DSS arose in the beginning of 70-s.

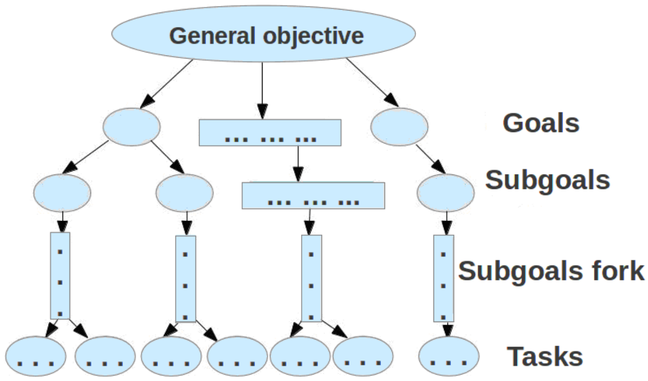


Fig. 4. Objectives tree.

The main stages of the process of FSN design are analysis of the requirements, formation of the objectives tree, development of the structural model of the FSN (structural synthesis), that is, definition of segments and subnetworks, topology design of segments and the network as a whole; development of FSN physical model, that is selection of equipment, protocols and methods of organization of data transmission channels; simulation modeling and FSN optimization.

On the basis of functional flow chart of the WSN construction process [4] in Fig. 3 we propose an extended functional flow chart of FSN architecture construction process, which can be the basis for a design support system of flying sensor networks.

Basing on received designer's requirements to an FSN and the plans on the network development, the objectives tree of FSN design is formed as well as criteria of achievement of the objectives and optimization parameters of the FSN design. A concept of the "objectives tree" was introduced by C. Churchman and R. Ackoff in 1957. An objectives tree is a structured, constructed on the hierarchy principle (distributed into levels, ranged) assembly of project objectives, in which the following ones are emphasized: the general objective ("tree root") and the subgoals of the first, second and consequent levels subject to it ("tree branches"). In Fig. 4 a generalized objectives tree is shown. In leaf nodes of the tree, simple tasks are formed. When designing communication networks, often the simple tasks are the requirements on achieving the specified thresholds of optimization parameters.

Examples of optimization parameters may be: probability of connection between F-nodes and the ICC; coefficient of network readiness; viability parameters (for example, number and fraction of aborted or functioning connections, mathematical expectation and the average fraction of number of died or survived nodes after a virtual attack on the edges or nodes); the total time of network functioning before the moment of its fault; amount of power consumed by the network/nodes in a fixed time interval; time to deliver messages from F-nodes to the ICC; time to deliver messages from the ICC to network nodes in any time moment; time of the network self-recovery after nodes faults; confidence coefficient for the data collected by the network; redundancy coefficient of the transmitted data; coverage area and the density of T-nodes allocation; network protection criteria (for example, network/nodes vulnerability to attacks, average time to dispose of the vulnerability, number of network vulnerabilities, criticality of attacking actions and threats); monetary cost of the network; monetary expenses for network allocation and operation etc.

The objectives tree of FSN design is constructed basing on methods of objectives structuring, knowledge bases about network structures, expert knowledge and expert systems, databases of templates of network structures design, databases of methods and algorithms of network structures optimization, specialized CAD systems and design support systems of FSNs.

For quality evaluation of the network design one can use a generalized indicator that characterizes (in percent) the degree of designer satisfaction of the designed network. One of the methods to produce the generalized indicator collecting in itself all parameters and requirements to the designed FSN is using fuzzy logic and applying confidence coefficients.

After having produced the objectives tree, one develops the assignment for FSN design and coordinates it with the designer. Next, the iteration process of constructing the flying sensor networks architecture is performed. The main steps of the process are given below.

1) Choose an information collection model to be used by the ground segment of the FSN. The functional block of choice of information collection model provides the designer with a list of available-to-use models of information collection together with their usage references. The database related to this block contains the full information about each information collection model (for example, the usage reference, software implementation, simulation model, a list of available-to-use routing algorithms and self-organization, restrictions on model applicability, compatibility with the models of interaction between the flying and the ground segments etc.). After having selected the certain information collection model, the designer uses the FSN design support system so as to set the concrete parameter values of the model. A promising approach is the automatic selection of parameter values of the model from the specified search space (in case the search space is very large and there are no concise algorithms or recommendations on selecting the model parameters, evolution, genetic, bio-inspired and other algorithms may be used to search for approximate solutions in a large space of variants).
2) Choose an information collection model to be used by the flying segment of the FSN. The functional block of this step implements choice of a compatible with Step 1 information collection model to be used by the flying segment of the FSN and operates in a similar way as the functional block of the first step.
3) Choose a model of interaction between the flying and the ground segments, compatible with the models chosen at the first two steps. The functional block of this step provides the designer with a list of available-to-use models of interaction between the segments together with their usage references. The database related to this block contains the full information about each model of interaction between the flying and the ground segments. After having selected the certain interaction model, the designer uses the FSN design support system so as to set the concrete parameter values of the model. As well as on Step 1, the module of automatic selection of model parameters can be used.
4) Choose the components of the ground segment of the FSN to be utilized. The functional block of choosing the FSN components implements the selection from the database of a set of inter-compatible types of utilized nodes on the current iteration. Selection of all compatible nodes is implemented with help of search with restrictions defined by the designer in the nodes database of the ground segment of the FSN. After that, a sorting of the set of compatible nodes is performed on the criteria set by the designer. The design support system of the FSN that implements the functional flowchart (Fig. 2) must either output a usage reference for the certain nodes set for the project, or provide the designer the ability to accomplish the choice themselves. The database related to this block contains the complete information about all available-to-use network nodes.
5) In accordance with the requirements to the FSN, allocate F-nodes at the information collection points and at the points from which it is required to direct the management actions on external objects.
6) Allocate the information collection centers and form the clusters of the ground structure of the FSN (for example Fig. 5). The functional block of segmentation into clusters forms the clusters basing on the clusterization algorithms available in the database, requirements for the

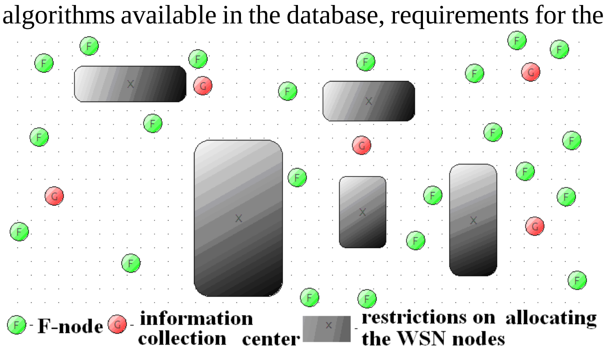


Fig. 5. Example of the information collection centers allocation.

network and the defined F-nodes allocation. It also performs the ICC allocation in such a way that each F-node would be able to connect to at least *K* ICCs. We suppose that information delivery from an ICC to an end user is performed without any losses or distortions and that the information received by the end user from any F-node is sufficient for them if it comes from at least one of the *K* ICCs with which the F-node is connected.

7) Choose a self-organization algorithm and a routing algorithm for the ground segment of the FSN (for example [5], [6]). The database related to this block contains the full information about each self-organization and routing algorithm and their simulation model. As well as on Step 1, the module of automatic selection of model parameters can be used.

8) Synthesize the structure of the ground segment of the FSN. The functional block of synthesis of the ground structure of the FSN (FB-SGS-FSN) allocates T-nodes in such a way that the synthesized network structure would satisfy design requirements and goals, that is, would have the “desired properties” for the designer. The initial data of the ground structure synthesis are the following: allocation ant type of F-nodes (Fig. 5); allocation and type of information collection centers (Fig. 5);

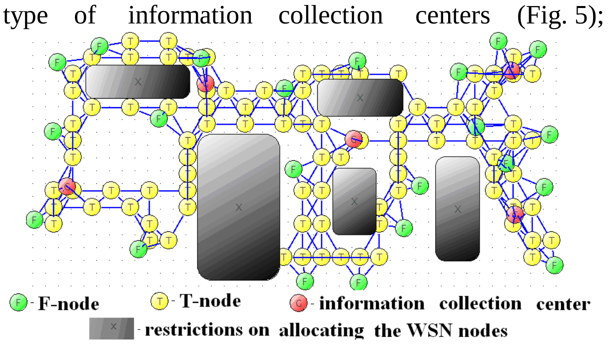


Fig. 6. Example of the designed fault-tolerant ground structure of the FSN (every F-node has at least 3 independent paths to at least 3 ICC)

description of the object at which the ground segment of the FSN is to be located at (the size of the object, its plan, spatial constraints on allocating T-nodes, barriers for electromagnetic waves expansion and characteristics of these barriers; types and characteristics of ready-to-use T-nodes; adopted self-organization and routing algorithms; the model of information collection and interaction between the flying and the ground segments; functional requirements; optimization parameters; the objective functions; exact and approximate functions to calculate network parameters; expert systems; simulation models etc. It is necessary to allocate T-nodes in such way (for example Fig. 6), that the designed ground segment of the FSN would have the «desired properties» assigned by a designer.

Example: Let $q = q(T)$ be the known probability of failure of one ICC in time *T*. Then the probability of at least one of *K* ICCs to be operative is calculated in the following manner: $P(Q)=1-q^K$. In case the requirement $P(Q)>= P^*$ is set, where P^* is a defined value, one can define the necessary ICCs number *K* solving the inequality $1-q^K>=P^*$ for *K*. In order to allow the information to be delivered from all F-nodes of the network to the end user upon failures of any $(K-1)$ ICCs, one needs to construct (with help of suitable T-nodes allocation) such WSN structure that would assure the defined minimal probability of each F-node to be connected with some *K* ICCs from the total number of *N*.

The FB-SGS-FSN in its operation process implements calculations of exact and approximate estimates of the structural network parameters, expert evaluation of intermediate results and simulation modeling of network operation. The results of the modeling and structural-parametric estimates of different parameters are passed onto input of the complex expert system of the ground network structure evaluation which is used to calculate the confidence coefficient K_{DSTR} of all designer's requirements for a ground structure of the FSN to be satisfied. In case K_{DSTR} does not satisfy designer's requirements, one should either continue synthesizing the ground structure of the FSN on Step 8, or go to selection of other parameters or models on different levels of the FSN construction (that is, to one of the steps 1-7).

In the paper [4] author proposes various bio-inspired algorithms and the functional flow chart of the multi-agent bio-inspired WSN structure design (Fig. 7).

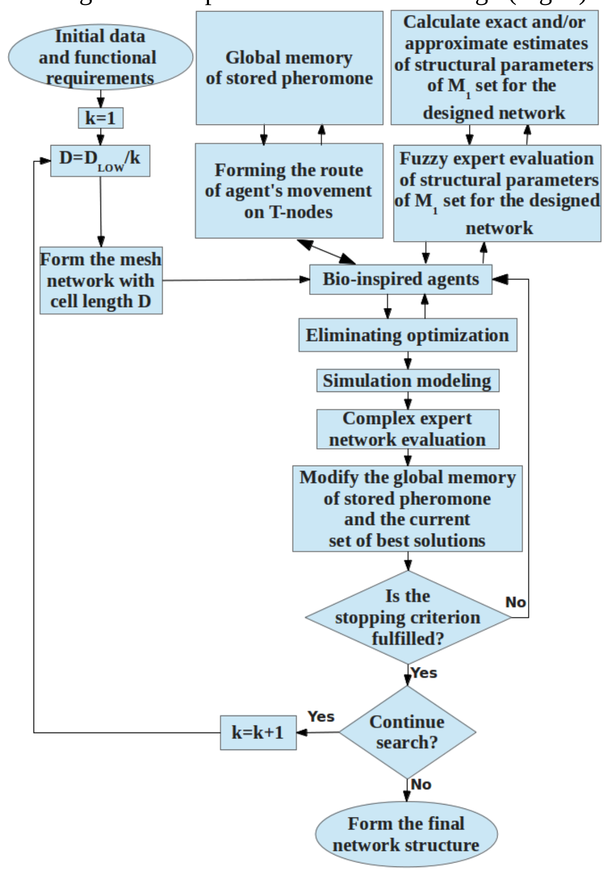


Fig. 7. Functional flow chart of the multi-agent bio-inspired WSN structure design [4].

In recent years, the research area of *Natural Computing* is rapidly developing. It unites mathematical methods in which the principles of natural mechanisms of decision making are embedded [7]. Scientists have developed bio-inspired algorithms (BA) modeling animals behavior for solving various optimization problems that either do not have exact solution, or the solutions search space is vary large and complex constraints of the objective function are present, as well as NP-complete.

The described recommendations on applying BA and the proof in [8] that even the constrained variant of the problem of minimal coverage on plane is NP-complete allow to conclude about the possibility to apply self-organizing bio-inspired algorithms for a self-organizing WSN structure design.

9) Choose the components of the flying segment of the FSN to be utilized. The functional block of choosing the flying segment components operates in a similar way as the functional block of the first step.
10) Choose a self-organization algorithm and a routing algorithm for the flying segment of the FSN (for example [9]). The functional block of the choice of the components of the flying segment operates in a similar way as the functional block of the Step 7.
11) Perform simulation modeling of the FSN operation as a whole. The modeling results and structural-parametric estimates of different parameters are passed onto input of the complex expert system of the FSN evaluation which is used to calculate the confidence coefficient K_{DALL} of all designer's requirements for the FSN architecture to be satisfied. In case K_{DALL} does not satisfy designer's requirements, one should either go to Step 9 or to Step 10.
12) Form the final design of the FSN architecture.

III. Conclusion

In this work we propose the functional scheme of the FSN architecture design, which can be the basis for a specialized design support system of flying sensor networks. Currently, the functional block of the FSN ground structure synthesis is implemented using *Java* programming language. Experimental research has shown the possibility of constructing the FSN ground structure considering various objective functions and optimization parameters.

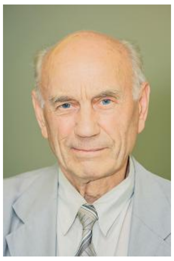
Acknowledgment

This work was supported by RFBR, research project No15 07-09431A “Development of the principles of construction and methods of self-organization for Flying Ubiquitous Sensor Networks”

References

[1] R. Kirichek, A. Paramonov, A. Koucheryavy. Flying Ubiquitous Sensor Networks as a Queueing System. *Proceedings, The IEEE 17th International Conference on Advanced Communication Technology, 2015. ICACT 2015*. Phoenix Park, Korea.
[2] A.E. Koucheryavy, A.G. Vladiko, R.V. Kirichek. Flying sensor networks. *Electrosvyaz*, vol.9, pp. 2-5, 2014.
[3] W O’Brien, C Julien, S Kabadayi, X Luo, J Hammer. An architecture for decision support in ad hoc sensor networks. *Electronic Journal of Information Technology in Construction* 14, 309-327
[4] V.A. Mochalov. Multi-agent Bio-inspired Algorithms for Wireless Sensor Network Design. *Proceedings, The IEEE 17th International Conference on Advanced Communication Technology, 2015. ICACT 2015*. Phoenix Park, Korea.
[5] Jn. Al-Karaki, AE Kamal. Routing techniques in wireless sensor networks: a survey. *Wireless communications*, IEEE 11 (6), 6-28.
[6] K. Sha, J. Gehlot, R. Greve. Multipath Routing Techniques in Wireless Sensor Networks: A Survey. *Wireless Personal Communications*. May 2013, Volume 70, Issue 2, pp 807-829.
[7] G. Rozenberg, T. Bäck, J.N. Kok. *Handbook of Natural Computing*, Springer, ISBN 978-3-540-92909-3, 2012.
[8] R.J. Fowler. “Optimal packing and covering in the plane are NP-complete”, *Inf. Process. Lett.*, vol.12(3), pp.133–137, 1981.
[9] S. Rosati, K. Kruzelecki, G. Heitz, D. Floreano, B. Rimoldi. Dynamic Routing for Flying Ad Hoc Networks. *IEEE Transactions on Vehicular Technology*. 06/2014.

Vladimir Mochalov was born in Lyubertsy, Russia in 1985. He received the Ph.D. degree in electronic engineering from Moscow Technical University of Communications and Informatics. His research interests include networks structure synthesis, artificial intelligence, bio-inspired algorithms, query answering systems and Big Data.



Anatoliy Pschenichnikov is a scientist, PhD, Professor, honorary worker of communications of the Russian Federation, head of "Communication Networks and Systems" department of Moscow technical university of communications and informatics. He is the author of over 200 scientific papers in the field of communications.

Volume 4 Issue 4, Jul. 2015, ISSN: 2288-0003

ICACT-TACT JOURNAL

Global IT
Research Institute

1713 Obelisk, 216 Seohyunno, Bundang-gu, Sungnam Kyunggi-do, Republic of Korea 463-824
Business Licence Number : 220-82-07506, Contact: secretariat@icact.org Tel: +82-70-4146-4991