INTELLIGENT MODEL OF USER BEHAVIOR IN DISTRIBUTED SYSTEMS

Andrii Shelestov, Serhiy Skakun, Olga Kussul

Abstract: We present a complex neural network model of user behavior in distributed systems. The model reflects both dynamical and statistical features of user behavior and consists of three components: on-line and off-line models and change detection module. On-line model reflects dynamical features by predicting user actions on the basis of previous ones. Off-line model is based on the analysis of statistical parameters of user behavior. In both cases neural networks are used to reveal uncharacteristic activity of users. Change detection module is intended for trends analysis in user behavior. The efficiency of complex model is verified on real data of users of Space Research Institute of NASU-NSAU.

Keywords: distributed systems, user behavior model, neural networks.

Introduction

At present the solution of complex large-scale problems arising in the areas of Earth observations from space [Shelestov, *et al.,* 2006], [Fusco, 2006], [Fusco, *et al.*, 2003], high-energy physics [Holtman, 2001], bioinformatics [Peltier, *et al.,* 2002], astronomy [Annis, *et al.,* 2002] is impossible without use of distributed computer systems. (e.g. Grid systems). Many tasks, such as computing and data resources sharing, distributed data processing, data storage, archiving and transfer are relied on them. One of the important challenges in the development of heterogeneous distributed infrastructure is the security provision. For this purpose many problems must be solved such as user authentication, authorization, rights delegation, etc. This can be done by using, for example, Globus Grid Security Infrastructure (GSI) [Foster, *et al.,* 1998] which is an extension of the Public Key Infrastructure [IETF], [Adams and Lloyd, 2002]. On the other hand, there are many monitoring tools intended for resources state and jobs monitoring (e.g. GridICE (http://gridice.forge.cnaf.infn.it/), MOGAS (http://ntu-cg.ntu.edu.sg/pragma/index.jsp)), but the do not provide monitoring of users' activities in order to detect anomalies and potential intrusions. Though, many sources report that the majority (80%) of information security incidents is perpetrated by insiders [Tulloch, 2003]. Hence, the problem of monitoring and detection of malicious user activity in distributed computer system is an important issue.

Related Works

Nowadays different methods and approaches are applied for the analysis of user activity. They are mostly based on the analysis and exposure of regularities and common actions in user behavior for automation, prediction, anomaly detection, etc. Data that are used for model construction possess individual features that define user behavior. For data processing different approaches can be applied such as history-matching methods and machine-learning methods.

In general, creation of user behavior model involves the following steps: data collection and data pre-processing, when useful information about user activity is collected from log-files; data processing, when feature extraction is done to represent data, and dimension reduction methods are used to reduce the size of the data; application of different techniques to obtain interesting characteristics of user behavior; interpretation of the results.

Among the existing approaches to user activity analysis we may consider so called Personal Security Programs that are used by commercial companies to monitor the activity of their employees. The results of such monitoring can be used to reveal malicious users in the case of information leakage, or to find out whether users use computers for their personal purposes. For example, such programs as PC Spy (www.softdd.com/pcspy/index.htm), Inlook Express (www.jungle-monkey.com), Paparazzi (www.industar.net) allow to capture and save screen images (screenshots) showing exactly what was being viewed by users. All screens can be captured, including Web pages, chat windows, email windows, and anything else shown on the monitor. However, these programs have some disadvantages; among them are high volume of stored information

and manual configuration of snapshots frequency. That is, if the frequency is low it would be rather difficult to find out something abnormal in user activity. Otherwise, a lot of the data should be stored.

Another example refers to Intrusion Detection Systems (IDS), particularly anomaly detection in computer systems. Usually, a model of normal user behavior is firstly created, so during monitoring any abnormal activity can be regarded as potential intrusion. Different approaches are applied to the development of anomaly detection systems: statistical methods [Javitz and Valdes, 1991], expert systems [Dowell and Ramstedt, 1990], finite automata [Kussul and Sokolov, 2003], neural networks [Cannady and Mahaffey, 1998], [Reznik, *et al.*, 1999], [Skakun, *et al.*, 2005], agent-based systems [Skakun, *et al.*, 2005], [Gorodetski, *et al.*, 2001], rule-based networks, genetic algorithms, etc.

It is worth mentioning that existing approaches do not provide adequate description of user behavior. There exist methods that exhibit only dynamical features of user behavior, and do not consider statistical properties, and vice versa. This paper describes a complex model of user behavior in distributed systems. The model consists of three components: on-line model, off-line model and change detection module. The use of on-line and off-line models allows the reflection of both dynamical and statistical features of user's activity. In order to provide adaptive and robust approach for the analysis and generalization of data obtained from user activity neural networks are applied. The proposed approach is verified on real data gathered during the work of users on the resources of Space Research Institute of NASU-NSAU.

Complex Model of Users Behavior in Distributed Systems

For adequate description of user behavior in distributed systems we propose complex model that consists of the following components:

- on-line model that describes user's activity during its work by predicting his actions;
- off-line model that is based on the analysis of statistical data acquired during user's work;
- change detection module intended for detection of trends in user's activity.

The proposed structure of complex model is depicted in Fig. 1.

For prediction of user actions neural network is used. The use of neural network is motivated by the fact that user behavior represents a complex non-linear process and by the need to reveal regularities in it. As a neural network paradigm, we use feed-forward neural network trained by means of back-propagation algorithm [Haykin, 1999]. Therefore, for each user a neural network is built, and trained in such way to predict user actions. The result of neural network work after completion of i-1 actions by user during session s_t is given by the following equation:

$$\hat{c}_i^{s_t} = F(\mathbf{x}_i),\ \mathbf{x}_i = \left(c_{i-1}^{s_t}, c_{i-2}^{s_t}, \ldots, c_{i-m}^{s_t}\right),$$

where F — non-linear transformation performed by neural network; $\mathbf{x}_i$, $\hat{c}_i^{s_t}$ — neural network input and output respectively; $c_i^{s_t}$ — the number of i-th user action during session s_t; m — number of previous actions used to predict the next one.

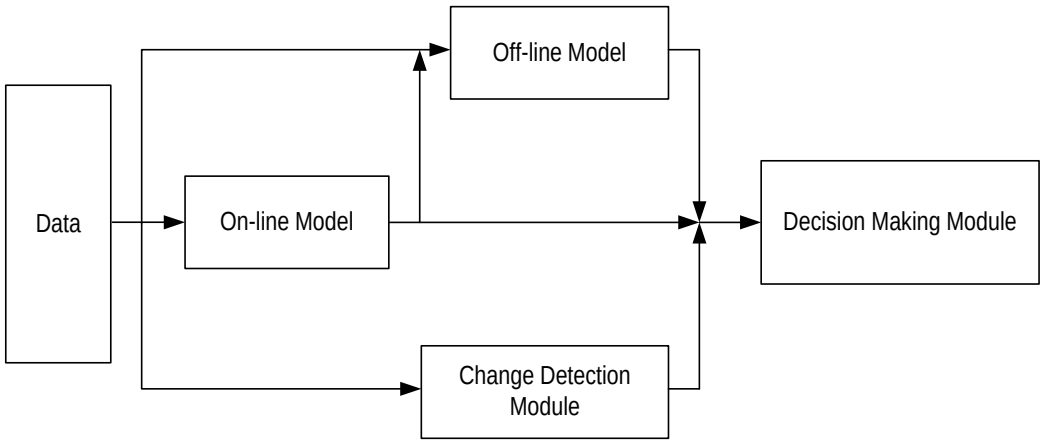


Fig. 1. Structure of complex neural network model of user behavior

On-line model describes dynamical features of user behavior by predicting user actions based on previous ones. The decision about user behavior (normal/abnormal) is based on the number of correctly predicted user actions by neural network (i.e. when $\hat{c}_i^{s_t}$ is equal to $c_i^{s_t}$). Construction of on-line model must also take into account the possibility that user behavior will be changing in the course of time. In order to provide adaptation of model to these changes (i.e. retraining neural network) change detection module is used.

On contrast, *off-line model* is based on the use of statistical (integral) parameters obtained during user behavior. The following set of characteristics about user behavior was taken:

$$\left\{n_{s_t}, o_{s_t}, h_{s_t}, d_{s_t}, s_{s_t}\right\}, \tag{1}$$

where n_{s_t} — number of actions performed by user; o_{s_t} — results of on-line model use, i.e. the number of correctly predicted user actions; h_{s_t} — user login host; d_{s_t} — user session duration time; s_{s_t} — the time of user session start.

This set is used as input feature to neural network for detection of normal/abnormal user activity. As in the case of on-line model, for each user feed-forward neural network is trained with back-propagation algorithm in order to distinguish normal and abnormal user behavior. The expected output of neural network during training is binary, i.e. 1 corresponds to normal behavior and 0 corresponds to anomaly. The neural network output is defined as follows:

$$\Delta_{s_t} = F(x_{s_t}), \quad \mathbf{x}_{s_t} = (n_{s_t}, o_{s_t}, h_{s_t}, d_{s_t}, s_{s_t}),$$

where F — non-linear transformation performed by neural network; $\mathbf{x}_{s_t}$, Δ_{s_t} — neural network input and output respectively; n_{s_t}, o_{s_t}, h_{s_t}, d_{s_t} s_{s_t} are defined by (1).

If an input to neural network is an independent sample (that was not used during training process), the corresponding output Δ_{s_t} will lie in the range [0; 1], and provide probability of user normal activity (higher values correspond to normal user behavior).

The user behavior does not represent stationary process, and it will be changing in the course of time (as a rule during 2-3 months). This can be caused by different reasons, e.g. due to software version changes, new tasks accomplishment. That is why, complex model is required to include *change detection module* that will detect trends in user behavior.

Let A be the alphabet of user actions, i.e. the set of all actions performed by user during set of sessions $\left\{s_t\right\}_{t=1}^{T}$. We assume that user behavior has not been changed during this time of work. Let N be the number of actions in alphabet A, and each action has a number from 1 to N. The number N+1 will be reserved to new actions that were not performed during sessions $\left\{s_t\right\}_{t=1}^{T}$. Let s_t (t>T) be the current session of user work with the following actions performed $\mathbf{c}^{s_t} = \left(c_1^{s_t}, c_2^{s_t}, \ldots, c_{N_{s_t}}^{s_t}\right)$. Then $c_i^{s_t} = N+1$, if $c_i^{s_t} \notin \{1, ..., N\}$. In order to detect changes in user behavior after session s_t we construct vector $\mathbf{g}(s_t)$ with the following components:

$$g_j(s_t) = \begin{cases} 1, \text{ if exists such } k = \overline{1,\ N_{s_t}}, \text{ that } c_k^{s_t} = j, \ j \in A \\ 0, \text{otherwise} \end{cases}.$$

That is, if an action was performed during session s_t, then corresponding component of vector $\mathbf{g}(s_t)$ is equal to 1, otherwise is equal to 0. Then the obtained vector $\mathbf{g}(s_t)$ is compared in pairs with vectors obtained during previous sessions s_{t-1}, s_{t-2}, …, s_{t-l}. As a measure of comparison Hamming distance is applied:

$$\aleph(\mathbf{g}(s_{t'}), \ \mathbf{g}(s_{t''})) = \sum_{j=1}^{N} \chi(g_j(s_{t'}), \ g_j(s_{t''})),$$

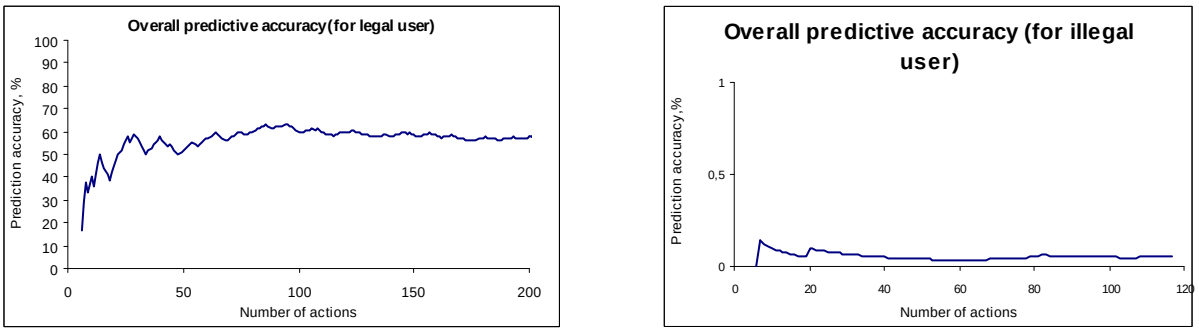
where $\chi(g_j(s_{t'}),\ g_j(s_{t''})) = \begin{cases} 1, & \text{if} \quad g_j(s_{t'}) \neq g_j(s_{t''}) \\ 0, & \text{otherwise} \end{cases}$. That is, $\aleph$ corresponds to number of components of two vectors that are different. As a result of comparisons we will obtain *l* values, following which we average and normalize on *N*: $H_t = \frac{1}{N}\left(\frac{1}{l}\sum_{k=1}^{l}\aleph(\mathbf{g}(s_t),\ \mathbf{g}(s_{t-k}))\right)$. If user behavior has not changed, then vector $\mathbf{g}(s_t)$ would not differ considerably from vectors for previous sessions. Hence, H_t would be below some threshold H_*: $H_t < H_*$. And vice versa, if anomaly occurred, vector $\mathbf{g}(s_t)$ would differ considerably from vectors for previous sessions and H_t would be under some threshold H^*: $H_t > H^*$. If $H_t \in (H^*; H^*)$, then a natural changes in user behavior took place.

Description of Experiments

Different experiments were run to demonstrate the efficiency of both on-line and off-line models and change detection module. For this purpose data needed for neural network training were acquired during a real work of users on the resources of Space Research Institute of NASU-NSAU.

Experiments for on-line model. For on-line model log files were transformed into format suitable for neural network. That is, for each user an alphabet of user actions was created, and each action was assigned an identifier (decimal number). For neural network input a binary coding was applied (7 bits per action). Feed-forward neural network was used to predict user actions based on 5 previous ones (the value of 5 previous actions was derived using autocorrelation function for sequence of user actions). Thus, the dimension of input data space was 35. In turn, for output data decimal coding was applied, and the dimension of output data space was 1. As to neural network architecture, we used neural network with 3 layers: input layer with 35 neurons, hidden layer with 35 neurons, and output layer with 1 neuron.

Then all data were randomly mixed and divided into train and test sets (70% for training and 30% for testing). Results of neural network work on test data showed that overall predictive accuracy (that is, the number of correctly predicted actions divided by total number) for different users varied from 33% to 59% (an example of overall predictive accuracy variations within number of actions is depicted in Fig. 2,a). To demonstrate that neural network was able to distinguish one user from another we run so called cross experiments. It was done in two ways. First one consisted in the following: data obtained during the work of one user (name him illegal user) were put to neural network trained for another (legal user). In such a case, overall predictive accuracy of neural network hardly exceeded 5% (it is shown on Fig. 2,b an example where overall predictive accuracy was 0,05%). Such experiment modeled the situation when illegal user logged on and begun to work under account of another user.



(a) *(b)*

Fig. 2. Predictive accuracy for: (a) legal user; (b) illegal user

The second method of cross experiments was done by inserting data of illegal user into data of legal one. This experiment modeled the situation when intruder begun to work under account of another user already logged on. In such a case, we used short-time predictive accuracy to measure the number of correctly predicted actions (this measure takes into account only last actions performed by user, for example, twenty last actions). Variations of short-time predictive accuracy for both legal and illegal user are depicted in Fig. 3. It is evident that short-time predictive accuracy for illegal user is considerably less then for legal one.

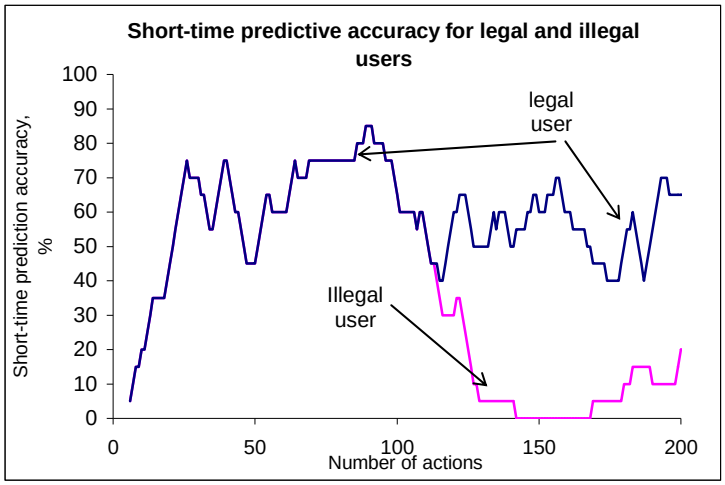


Fig. 3. Short-time predictive accuracy

Experiments for off-line model. Considering off-line model, all needed statistical data were obtained from log files. Then they were encoded, divided into training and test sets, and input to neural network. Results of neural network work on test data gave 80% accuracy of correct user behavior identification. That is, experiments showed that off-line model was able to distinguish normal and abnormal (anomalous) user behavior.

Experiments for change detection module. In order to verify change detection module first H_t was estimated for those user sessions when the use behavior was normal. Derived values of H_t did not differ considerably and lied in the range (0; H_*). (For each user different values of threshold H_* were obtained and varied from 0,1 to 0,17). For anomaly behaviour modelling cross-experiments were run. That is, vector $\mathbf{g}(s_t)$ was calculated for sessions of another user and compared with vectors obtained for a given user. Obtained values of H_t considerably increased (average in 2,5 times).

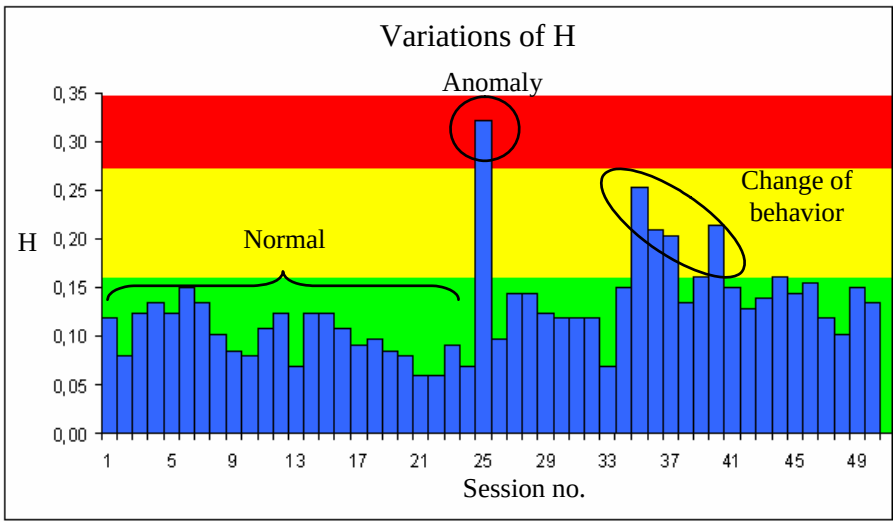


Fig. 4. Variations of value H_t depending on user behavior

In order to model natural changes in user behavior (that do not correspond to anomaly) for each user components of vector $\mathbf{g}(s_t)$ were randomly changed (with probability 0,25). In this case the value H_t increased in 2 times, and then decreased to ordinary level.

The variations of value H_t for typical user depending on user behavior are depicted in Fig. 4. Sessions #1-24 correspond to normal behavior, and H_t lie in the range (0; 0,15). When data of another user were inserted

(session #25) H_t increased up to 0,32. When we modeled natural change of user behavior (sessions #35-50) H_t increased up to 0,25, and then decreased to ordinary level.

Therefore, experimental results showed the possibility of proposed complex neural network model to distinguish with confidence normal and abnormal (anomalous) user behavior.

Conclusions

In this paper we proposed a complex model of user behavior in distributed system. In order to adequately describe different features of user behavior the model consists of three components: on-line model considers dynamics of user behavior by predicting user actions; off-line model is based on the analysis of statistical parameters, and change detection module that is intended for detection of trends in user's activity. In contrast to existing methods the proposed model enables complex analysis of user behavior both during its work (in real-time) and after user's work completion (in off-line mode). The use of neural network provides intelligent approach to analysis and generalization of data acquired during user activity. In order to demonstrate efficiency of complex model different experiments were run on real data obtained during user work on resources of Space Research Institute of NASU-NSAU. The results of experiments showed applicability of the proposed approach.

Acknowledgement

This research is supported by INTAS-CNES-NSAU project "Data Fusion Grid Infrastructure", Ref. No 06-1000024-9154.

Bibliography

[Adams and Lloyd, 2002] Adams C., Lloyd S. "Understanding PKI: Concepts, Standards, and Deployment Considerations", 2nd ed. Addison-Wesley, 2002.

[Annis, *et al.*, 2002] Annis J., Zhao Y., et al., "Applying Chimera Virtual Data Concepts to Cluster Finding in the Sloan Sky Survey," Technical Report GriPhyN-2002-05, 2002.

[Cannady and Mahaffey, 1998] Cannady J., Mahaffey J. "The Application of Artificial Neural Networks to Misuse Detection: Initial Results", In Proc. of the 1998 National Information Systems Security Conf. (NISSC'98), Arlington, VA, 1998.

[Dowell and Ramstedt, 1990] Dowell C., Ramstedt P. "The ComputerWatch data reduction tool", In Proc. 13th National Computer Security Conf., 1990, pp. 99–108.

[Foster, *et al.*, 1998] Foster I., Kesselman C., Tsudik G., Tuecke S. "A Security Architecture for Computational Grids", In ACM Conf. on Computers and Security, 1998, pp. 83-91.

[Fusco, 2006] Fusco L. "Earth Science GRID on Demand", Presented on CEOS WGISS-21 GRID Task Team meeting, Budapest, Hungary, May 2006.

[Fusco, *et al.*, 2003] Fusco L., Goncalves P., Linford J., Fulcoli M., Terracina A., D'Acunzo G. "Putting Earth-Observation on the Grid", ESA Bulletin, 2003, 114, pp. 86-91.

[Gorodetski, *et al.*, 2001] Gorodetski V., Karsaev O., Khabalov A., Kotenko I., Popyack L., Skormin V. "Agent-based model of Computer Network Security System: A Case Study", In Proc. of the Int. Workshop 'Mathematical Methods, Models and Architectures for Computer Network Security', Lecture Notes in Computer Science, 2052, Springer Verlag, 2001, pp. 39-50.

[Haykin, 1999] Haykin S. "Neural Networks: a comprehensive foundation", Upper Saddle River, New Jersey: Prentice Hall, 1999, 842 p.

[Holtman, 2001] Holtman K., "CMS Requirements for the Grid", In Proc. of the Int. Conf. on Computing in High Energy and Nuclear Physics (CHEP2001), 2001.

[IETF] IETF, Public-Key Infrastructure (pkix) Charter.

[Javitz and Valdes, 1991] Javitz H., Valdes A. "The SRI IDES statistical anomaly detector", In: Proc. IEEE Symp. on Research in Security and Privacy, 1991, pp. 316–326.

[Kussul and Sokolov, 2003] Kussul N., Sokolov A. "Adaptive anomaly detection of user behaviour using Markov chains with variable order", J. of Automation and Control, Vol. 4, pp. 83-88. (in Russian)

[Peltier, *et al.*, 2002] Peltier S.T., et al. "The Telescience Portal for Advanced Tomography Applications", J. of Parallel and Distributed Computing: Computational Grid, 2002, 63(5), pp. 539-550.

[Reznik, *et al.*, 1999] Reznik A., Kussul N., Sokolov A. "Identification of user activity using neural networks", J. of Cybernetics and Computer Science, 1999, No. 123, pp. 70–79. (in Russian)

[Ryan, *et al.*, 1998] Ryan J., Lin M-J., Miikkulainen R. "Intrusion Detection with Neural Networks", Advances in Neural Information Processing Systems, Cambridge, MA: MIT Press, 1998, pp. 943–949.

[Shelestov, et al., 2006] Shelestov A.Yu., Kussul N.N., Skakun S.V. "Grid Technologies in Monitoring Systems Based on Satellite Data", J. of Automation and Information Science, 2006, Vol. 38, Issue 3, pp. 69-80.

[Skakun, *et al.*, 2005] Skakun S.V., Kussul N.N., Lobunets A.G. "Implementation of the Neural Network Model of Users of Computer Systems on the Basis of Agent Technology", J. of Automation and Information Sciences, 2005, Vol. 37, Issue 4, pp. 11-18.

[Tulloch, 2003] Tulloch M. "Microsoft Encyclopedia of Security", Redmond, Washington: Microsoft Press, 2003, 414 p.

Authors' Information

Andrii Yu. Shelestov – PhD, Senior Researcher, Department of Space Information Technologies and Systems, Space Research Institute of NASU-NSAU, Glushkov Ave 40, Kyiv-187, 03650 Ukraine, e-mail: inform@ikd.kiev.ua.

Serhiy V. Skakun – PhD, Research Assistant, Department of Space Information Technologies and Systems, Space Research Institute of NASU-NSAU, Glushkov Ave 40, Kyiv-187, 03650 Ukraine, e-mail: inform@ikd.kiev.ua.

Olga M. Kussul – BSc, Physics and Technology Institute, National Technical University "KPI", Peremoga Ave 37, Kyiv-056, 03056 Ukraine, e-mail: olgakussul@gmail.com.

DATA ASSIMILATION TECHNIQUE FOR FLOOD MONITORING AND PREDICTION

Natalia Kussul, Andrii Shelestov, Serhiy Skakun, Oleksii Kravchenko

Abstract. This paper focuses on the development of methods and cascade of models for flood monitoring and forecasting and its implementation in Grid environment. The processing of satellite data for flood extent mapping is done using neural networks. For flood forecasting we use cascade of models: regional numerical weather prediction (NWP) model, hydrological model and hydraulic model. Implementation of developed methods and models in the Grid infrastructure and related projects are discussed.

Keywords: Grid computing, remote sensing, modeling, international Grid projects.

Introduction

Nowadays Grid represents a powerful technology for solution of complex large-scale problems arising in such areas as high-energy physics [1], gravitational-wave physics [2], astronomy [3], bioinformatics [4], Earth Observations (EO) [5, 6, 7], etc. To this end the efficiency of application of Grids in different domains was demonstrated in numerous projects: EGEE [8], GriPhyN [9], DataGrid [10], CrossGrid [11] etc. The advantages of Grids come from its ability to integrate heterogeneous computational and informational resources managed by different distributed organizations [12]. It is particularly important for EO domain where one needs to manage with large amounts of data acquired from different satellites in different spectral bands that need to be integrated with aerial and in-situ components and maps; complex workflows; distributed archives and so on [5, 7].

Today remote sensing data from space are widely used for natural hazards and environmental monitoring, land use management, agriculture, etc. Floods are among the most devastating natural hazards in the world, affecting more people and causing more property damage than any other natural phenomena [13]. The dramatic floods of Central and Eastern Europe in August 2002 and Spring 2001 and 2006 emphasize the extreme in climatic variations. Ukraine is also vulnerable to floods as, in particular in the Carpathian region where it occurs almost