UDC - 004.725:004.852

APPLICATION OF A GENERATIVE-ADVERSARIAL NETWORK FOR MANAGING A PRECISE STOCHASTICALLY CHANGEABLE SIGNAL SOURCES

Timur V. Jamgharyan
National Polytechnical University of Armenia
105 Teryan St., Yerevan, RA
t.jamgharyan@yandex.ru
ORCID iD: 0000-0002-9661-1468
Republic of Armenia

Abstract

The paper presents the results of calculations and tests of stochastically changeable sources of precise time in a network infrastructure. The simulation of the attack of availability for stochastically changeable sources of precise time signals managed by a generative-adversarial network has been carried out. To synchronize the decentralized network infrastructure, a minimum number of stochastically changing precise time signals sources has been determined.

Key words: regression, precise time signals, generative-adversarial network, attack of availability.

Introduction

Various devices («СТВ-01, ИСС-1.3», SEL-2488, PTP-1U) are used to provide data transmission networks with precise time (PT) signals the protocols of which are described in the scientific papers [1, 2, 3, 4]. With the growth of network-centric conflicts and network attacks based on M2M (Machine-to-Machine, M2M), the number of types of attacks both on sources of PT and on the network infrastructure (NI) itself, where the source of the exact time is the generator of the attack is also growing [5, 6].

When using highly dynamic network devices that constantly change their geospatial position (a swarm of unmanned aerial vehicles, many mobile terminals), the task of synchronizing all devices arises. The use of a stationary PT source is not always advisable, since in the event of an attack, the stationary PT source will be one of the primary targets. To ensure the stable functioning of the network infrastructure, the PTS source must move within the network infrastructure in accordance with the stochastic law. This problem is being actively investigated [7, 8, 9, 10, 11, 12], but the proposed solutions have certain limitations.

In particular, it is proposed to increase the number of PT sources during their pseudo-random movement, the secret of which lies in the movement algorithm. But this method has several disadvantages:

- all synchronized devices must know the switching order (key),
- with long-term observation, you can open the key to the switching algorithm and, as a result, attack the next PT source,
- impossibility of introducing new devices into the network infrastructure that do not have a switching algorithm,
- in any case, the system is centralized, since the number of PT sources is finite and small in relation to the entire set of synchronized devices.

To overcome these limitations, the model uses machine learning (ML) and deep learning (DL) techniques.

These terms mean: machine learning - is a class of artificial intelligence methods in which a solution to a problem is not found directly, but by using solutions to many similar problems, and deep learning is a set of machine learning methods based on the study of representations [13, 14].

Based on the above mentioned, the development of an algorithm for stochastic movement of sources of precise time signals in the network infrastructure becomes relevant. The scientific novelty of the research lies in the study of a model of stochastically moving sources of PT signals within the existing constraints.

Conflict setting and set of methodology

Development of an algorithm for stochastic switching of sources of PT signals in highly dynamic network infrastructures.

Three tasks are solved within the framework of the study:

1. determination of the boundary conditions of the algorithm,
2. development of an algorithm for switching PT sources to the state of a synchronized device and vice versa,
3. development of software that allows connecting synchronized devices to STV sources on the basis of a generative adversarial network [15].

Stage 1. Boundary conditions of the developed algorithm.

1. all devices must be both a PT source (server, S) and a synchronized device (client, C),
2. sampling from several PT sources must satisfy the conditions of the Marzullo algorithm (an algorithm for selecting a PT source from several sources of different accuracy) [16],
3. the probability of a geo-positional discrepancy between the servers and clients of the CTB should not exceed the value of the Kullback-Leibler interval (at the physical level, this limitation means that there should be communication between devices) [17].

Stage 2. Development of an algorithm for stochastic switching from the server state to condition of the client.

As a tool for the implementation of the second stage, a generative adversarial network has been chosen, which allows stochastic switching of the device from the state of the source of precise time signals to the state of the synchronized device. As a method for switching the state of devices, a «supervised learning» generative learning model («supervised learning» model is an algorithm, the prediction of which is based on existing templates) [18, 19] was chosen.

The choice of that method is conditional on the fact that the set of data of the functioning NI is known.

To determine the learning rate of a GAN, equation (1) is used to obtain [16].

$$a_t = a_{start} - (a_{start} - a_{end}) \times \frac{t}{K} \quad (1)$$

where a_t - learning rate, t - learning step, a_{start} – initial learning rate, a_{end} - end learning rate, K - number of iterations.

The learning rate of a generative adversarial network is a parameter at which the network learns to approximate the input-output relationship that is contained in the training data. The

manifestation of learning is a change in weight and a change in the rate of change in weight which determines the rate of learning [21, 22].

To determine the probability of failure of the decentralized PT system, equation (2) is used to obtain [17]

$$P(A) = \frac{(S-s+1)\times(S-s+2)\times\ldots\times(S-s+N)}{(S+1)\times(S+2)\times\ldots\times(S+N)} \quad (2)$$

where

P(A) - the probability of NI destabilization, *N* - the total number of PT servers and clients, *S* - the number of all PT servers, *s*-- the number of PT servers selected by the synchronized devices at a given time.

The use of equation (2) is due to the fact that a stochastically switched system of sources of precise time signals is decentralized due to the absence of a central stratum. A scheme is also implemented in distributed registry systems where each node (in our case, a synchronized device) randomly selects N_{nodes} to replenish and verify its database (in this case, update the time).

On the basis of equations (1.2), the equation (3) was obtained which determines the critical value of the number of sources of PT, above which the system becomes desynchronized. This inequality was obtained based on the solutions described in [13,24,25,26] in particular, the concept of logits is considered in [13, 24, 25] and the method for determining the coefficient of learning rate and decisions based on interactive the proofs are considered in [26].

$$P(A)_{a_t} \leq K_{crtttcat} \quad (3)$$

where, $P(A)_{a_t}$ is probability of desynchronization of the NI from retraining of the GAN, $K_{crtttcat}$ is the critical value of a decentralized synchronization system.

The critical number of PT sources is understood to be such a number at which an increase by one source leads to «overfitting» of the generative adversarial network.

Experimental procedures

1. Description of *GAN* working

The developed GAN based on previously entered data sets begins the procedure of stochastic switching server PT to client PT and client PT to server PT. Any PT client on the basis of the boundary conditions «searches» for PT servers and when conditions 2, 4 of stage 1 are satisfied, connects to it updating its time base. The training used a GAN with equal weights. The software implementation of the generative adversarial network was carried out using the «scikit-learn» library [27, 28, 29, 30]. Fig. 1 shows a model of a neural network of stochastic switching between PT servers and PT clients based on GAN with forward signal propagation.

2. Description of practical modeling

In the virtual environment of Microsoft Hyper-V Server 2016 [31], a virtual software-defined network is deployed based on 100 virtual routers (Cloud hosted router, CHR, based on Mikrotik license) [32] with installed NTP (Network time protocol, NTP) server packages and NTP client.

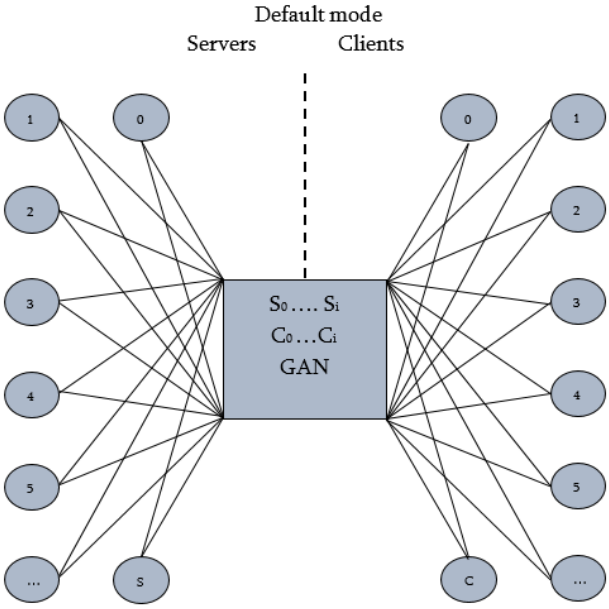

Fig. 1 A model of neural network of stochastic switching between CTB servers and PT clients based on GAN with direct signal propagation

S_0-S_i- - time server datasets, C_0-C_i - time clients datasets,GAN - generative adversarial network.

The purpose of virtual routers is to model a virtual network infrastructure. Various configurations are configured in the virtual NI. The procedure for turning on/off routers, switching to the NTP server or NTP client mode, changing the routing parameters is carried out using the developed software. The training of the generative network was carried out in 8 stages, 22 iterations each (at each stage, data sets of different levels of the OSI model were entered, the behavior of the virtual network was measured, a parameter was determined that led to a change in the source of the exact time signals). The hardware parameters of the server in which the measurements were carried out are presented in Tab. 1. The block diagram of the virtual environment is shown in Fig. 2.

Table 1

Type	Total RAM	CPU	Operation System	RAM for one CHR (Server/Client)
Dell Power Edge T-330	128 Gb	Intel Xeon E3-1200 v6	Microsoft Hyper-V Server 2016	1Gb

where, RAM- random access memory, CPU-central processor unit.

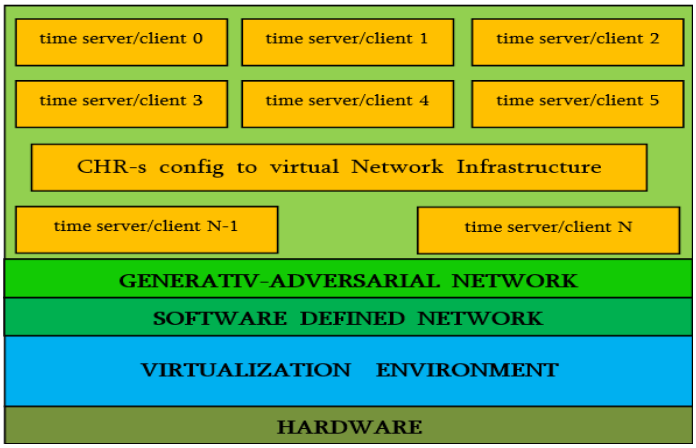

Fig. 2 The block diagram of the virtual environment

Research results

For training the generative adversarial network and calculations, a test dataset is used which is obtained from the existing data transmission network as well as three peering exchange points.

The results of studying a network infrastructure consisting of 100 devices controlled by a generative adversarial network are presented in Tab. 2, with $0{,}43 < K_{crittcal} \leq 0{,}45$.

The calculations were performed in the following software environments:

1. IBM SPSS Statistics [33],
2. Matlab (Statistics and Mashine learning toolbox) [34].

Table 2

t	MAE Actual	MAE Predicted	RMSE Actual	RMSE Predicted	a_t Actual	a_t Predicted
0	3,485	3,496	1,185	1,218	8	3* 7** 10***
1	4,128	4,278	1,431	1,476	14	5* 9** 11***
2	5,527	5,573	1,536	1,542	17	6* 9** 13***
3	6,134	6,146	1,647	1,680	19	8* 11** 14***
4	7,528	7,614	1,852	1,873	22	9* 16** 20***

where,

t is learning step, a_t is learning rate, MAE mean absolute error, RMS root mean squared error, * - undertrained generative-adversarial network, ** - trained generative-adversarial network, *** - retrained generative-adversarial network (overfitiing GAN).

Figures 3, 4, 5 show graphs of predicted (P) and actual (A) values of linear regression for undertrained, trained and retrained (overfitting) GAN.

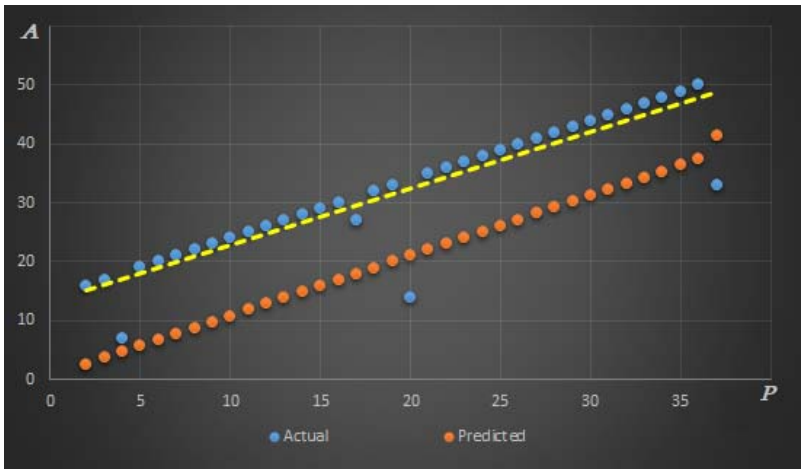

Fig. 3 The values of linear regression for undertrained GAN

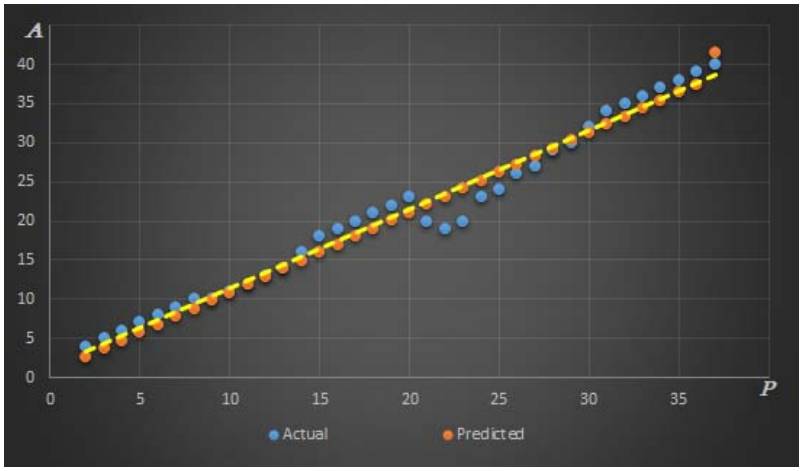

Fig. 4 The values of linear regression for trained GAN

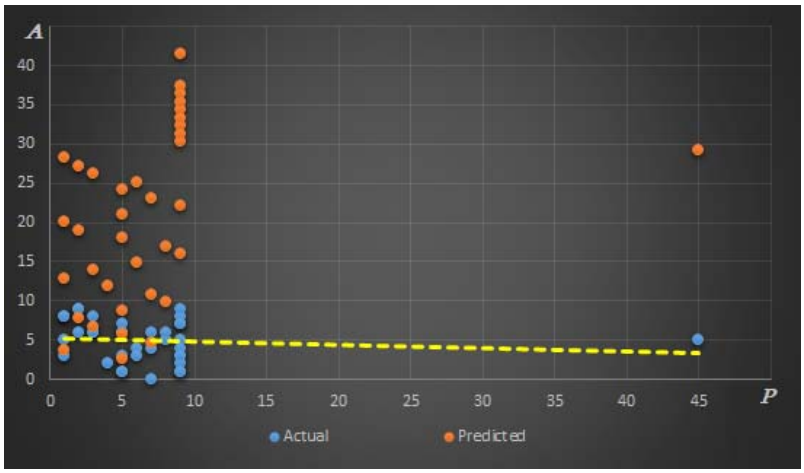

Fig. 5 The value of linear regression for retrained GAN

Conclusion

The research of the use of a GAN for the control of stochastically moving sources of precise time signals has been carried out. A model and software have been developed that allow connecting synchronized devices to sources of precise time signals based on a generative adversarial network. The results obtained confirm that with a trained generative adversarial network, it is possible to create a network infrastructure with a decentralized stochastically moving source of precise time signals.

References

1. Network Time Protocol Version 4: Protocol and Algorithms Specification (2010) //rfc 5905.- p. 110.
2. Precision Time Protocol Version 2 (PTPv2). Management Information Base (2017) //rfc 8173.- p.64.
3. Synchronization @Mobile networks (2016) //ALBEDO telecom.- p.15.
4. Stratum One Time Servers. URL: http://support.ntp.org/bin/view/Servers/StratumOneTimeServers
5. NTP Amplification Attcks.E-2013-5211.Cubersecurity&Infrastructure Security Agensy (2016) // URL: https://us-cert.cisa.gov/ncas/alerts/TA14-013A.
6. Kopp D., Dietzel C., Hohfeld O. DDoS Never Dies? An IXP Perspective on DDoS Amplifacation Attacks (2021) //MPI for Informatics, Brandenbirg University of Technology, URL: https://arxiv.org/abs/2103.04443.
7. Annessi R., Fabini J., Zseby T. Secure Time: Secure Multicast Time Synchronization (2017) //Institute of Telecommunications, TU Wien, Austria,

URL: https://arxiv.org/ftp/arxiv/papers/1705/1705.10669.pdf

8. Narula L., Humpphreys T. Requirements for Secure Clock Synchronization (2018) //Univercity of Texas at Austin, URL: https://arxiv.org/abs/1710.05798.
9. Tripathi N., Hubballi N. Preventiog Time Synchronization in NTP`s Broadcast Mode (2020) //Techical Univercity of Darmstadt Germany (N.Tripathi), Indian Institute of Technology Indore, India (N.Hubballi), URL: https://arxiv.org/pdf/2005.01783.pdf
10. Jejtner P., Shulman H., Waidner M. Pitfalls of Provably Secure Systems in Internet. The Case of Chronos NTP (2020) //Technical Univercity of Darmstadt, Fraunhofer Institute for Secure Information Technology, URL: https://arxiv.org/abs/2010.08460.
11. Mtibaa A., Mastorakis S. NDNTP: A Named Data Networking Time Protocol (2020) //University of Missouri-St.Louis, US (A.Mtibaa), Iniversity of Nebraska, Omaha, US (S Mastorakis), URL: https://arxiv.org/pdf/2007.07807.pdf
12. Obleukhov O., Byagowi A. Open-sourcing a more precise time appliance (2021) //URL: https://engineering.fb.com/2021/08/11/open-source/time-appliance/.
13. Buduma N., Locastio N. Fundamentals of Deep Learning. Designing Next-Geberation Mashine Intelligense Algorithms (2020) // Beijing-Boston-Farnham-Sebastopol-Tokyo, O`REILLY®.- p. 306
14. Novothny J., Bilokon P., Galiotos A., Deleze F. Machine Learning and Big Data with kdb+/q, (2020) // Wiley.- p. 614.
15. Goodfellow I.J., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A., Bengio Y. Generative Adversarial Networks (2014) //University de Montreal, URL: https://arxiv.org/abs/1406.2661
16. Marzullo K.A. Maintaining the Time in a Distributed System: An Example of a Loosely-Coupled Distributed Service (1984) //Ph.D. dissertation, Stanford University, URL: https://www.researchgate.net/publication/221344156_Maintaining_the_Time_in_a_Distributed_System
17. Bobkov S.G. The proximity of probability distributions in terms of Fourier-Stieltjes transforms, Uspekhi Mat.Nauk (2016) //volume 71, issue 6 (432).- p. 37-98.
18. Ng A., Soo K. Data Sciense for the Layman (2019) //Programmer's Library, Peter.- p. 208.
19. Skansi S. Guide to Deep Learning Basics. Logical, Historical and Philosopial Perspectives (2020) //Univercity of Zagreb, Publishing Springer.- p. 144.
20. Weidman S. Deep Learning from Scratch. Building with Python from First Principes (2019) //O`REILLY®.- p. 245.
21. https://radioprog.ru/post/773
22. Fenner M.E. Machine learning with Python for Everyone (2020) //Publishing Addison-Wesley.- p.588.
23. Kolesnikov P., Beketova Y., Krylov G. Blockchain Technology: Attack Analysis, Defense Strategies (2017) //Academic Publishing LAMBERT.- p. 77.
24. https://wiki.loginom.ru/articles/learning-rate.html
25. Hapke H., Nelson C. Building Machine Learning Pipelines. Automatic Model life Cecles with TensorFlow (2021) //Beijing-Boston-Farnham-Sebastopol-Tokyo, O`REILLY®.- p. 346.
26. Zapechnikov S.V. Cryptographic protocols. Probabilistic evidence, course of lectures (2018) //National Research Nuclear University.- p. 22.
27. Scikit-learn. Machine Learning in Python // URL: https://scikit-learn.org/stable/index.html
28. Navlani A., Fandango A., Idris I. Python Data Analysis (2021) //Birminham-Mumbai, Packt.- p.463.
29. Ravichandiran S. Deep Reinforcement Learning with Python (2020) //Birminham-Mumbai, Packt.- p. 761.

30. Foster D. Generative Deep Learning. Teaching Machines to Paint, write, Compose and Play (2019) //Beijing-Boston-Farnham-Sebastopol-Tokyo, O`REILLY®.- p. 330.
31. https://www.microsoft.com/ru-ru/evalcenter/evaluate-hyper-v-server-2016
32. Mikrotik official Web-site // URL: https://mikrotik.com/
33. IBM corporated official website software section: [Online].Available.https://www.ibm.com/analytics/spss-statistics-software
34. https://www.mathworks.com/

References

1. Network Time Protocol Version 4: Protocol and Algorithms Specification (2010) //rfc 5905.- p. 110.
2. Precision Time Protocol Version 2 (PTPv2). Management Information Base (2017) //rfc 8173.- p.64.
3. Synchronization @Mobile networks (2016) //ALBEDO telecom.- p.15.
4. Stratum One Time Servers. URL: http://support.ntp.org/bin/view/Servers/StratumOneTimeServers
5. NTP Amplification Attcks.E-2013-5211.Cubersecurity&Infrastructure Security Agensy (2016) // URL: https://us-cert.cisa.gov/ncas/alerts/TA14-013A.
6. Kopp D., Dietzel C., Hohfeld O. DDoS Never Dies? An IXP Perspective on DDoS Amplifacation Attacks (2021) //MPI for Informatics, Brandenbirg University of Technology, URL: https://arxiv.org/abs/2103.04443.
7. Annessi R., Fabini J., Zseby T. Secure Time: Secure Multicast Time Synchronization (2017) //Institute of Telecommunications, TU Wien, Austria,
   URL: https://arxiv.org/ftp/arxiv/papers/1705/1705.10669.pdf
8. Narula L., Humpphreys T. Requirements for Secure Clock Synchronization (2018) //Univercity of Texas at Austin, URL: https://arxiv.org/abs/1710.05798.
9. Tripathi N., Hubballi N. Preventiog Time Synchronization in NTP`s Broadcast Mode (2020) //Techical Univercity of Darmstadt Germany (N.Tripathi), Indian Institute of Technology Indore, India (N.Hubballi), URL: https://arxiv.org/pdf/2005.01783.pdf
10. Jejtner P., Shulman H., Waidner M. Pitfalls of Provably Secure Systems in Internet. The Case of Chronos NTP (2020) //Technical Univercity of Darmstadt, Fraunhofer Institute for Secure Information Technology, URL: https://arxiv.org/abs/2010.08460.
11. Mtibaa A., Mastorakis S. NDNTP: A Named Data Networking Time Protocol (2020) //University of Missouri-St.Louis, US (A.Mtibaa), Iniversity of Nebraska, Omaha, US (S Mastorakis),
    URL: https://arxiv.org/pdf/2007.07807.pdf
12. Obleukhov O., Byagowi A. Open-sourcing a more precise time appliance (2021) //URL: https://engineering.fb.com/2021/08/11/open-source/time-appliance/.
13. Будума Н., Локашо Н. Основы глубокого обучения.Создание алгоритмов искусственного интеллекта следующего поколения (2020) // «Манн, Иванов и Фербер» Москва, Beijing-Boston-Farnham-Sebastopol-Tokyo, O`REILLY®.- с. 306.
14. Novothny J., Bilokon P., Galiotos A., Deleze F. Machine Learning and Big Data with kdb+/q, (2020) // Wiley.- p. 614.
15. Goodfellow I.J., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A., Bengio Y. Generative Adversarial Networks (2014) //University de Montreal, URL: https://arxiv.org/abs/1406.2661
16. Marzullo K.A. Maintaining the Time in a Distributed System: An Example of a Loosely-Coupled Distributed Service (1984) //Ph.D. dissertation, Stanford University, URL:

https://www.researchgate.net/publication/221344156_Maintaining_the_Time_in_a_Distributed_System

17. Бобков С.Г. Близость вероятностных распределений в терминах преобразований Фурье-Стилтьеса, УМН (2016) //том 71, выпуск 6(432).- с. 37-98.
18. Ын А., Су К. Теоретический минимум по Big Data. Все что нужно занть о больших данных (2019) //Библиотека программиста, изд, Питер.- с. 208.
19. Skansi S. Guide to Deep Learning Basics. Logical, Historical and Philosopial Perspectives (2020) //Univercity of Zagreb, Publishing Springer.- p. 144.
20. Weidman S. Deep Learning from Scratch. Building with Python from First Principes (2019) //O`REILLY®.- p. 245.
21. https://radioprog.ru/post/773
22. Fenner M.E. Machine learning with Python for Everyone (2020) //Publishing Addison-Wesley.- p.588.
23. Колесников П., Бекетова Ю., Крылов Г. Технология блокчейн.Анализ атак, стратегии защиты (2017) //Academic Publishing LAMBERT.- с. 77.
24. https://wiki.loginom.ru/articles/learning-rate.html
25. Hapke H., Nelson C. Building Machine Learning Pipelines. Automatic Model life Cecles with TensorFlow (2021) //Beijing-Boston-Farnham-Sebastopol-Tokyo, O`REILLY®.- p. 346.
26. Запечников С.В. Криптографические протоколы. Вероятностные доказательства, курс лекций (2018) //Национальный исследовательский ядерный университет «МИФИ».- с. 22.
27. Scikit-learn. Machine Learning in Python // URL: https://scikit-learn.org/stable/index.html
28. Navlani A., Fandango A., Idris I. Python Data Analysis (2021) //Birminham-Mumbai, Packt.- p.463.
29. Ravichandiran S. Deep Reinforcement Learning with Python (2020) //Birminham-Mumbai, Packt.- p. 761.
30. Foster D. Generative Deep Learning. Teaching Machines to Paint, write, Compose and Play (2019) //Beijing-Boston-Farnham-Sebastopol-Tokyo, O`REILLY®.- p. 330.
31. https://www.microsoft.com/ru-ru/evalcenter/evaluate-hyper-v-server-2016
32. Mikrotik official Web-site // URL: https://mikrotik.com/
33. IBM corporated official website software section: [Online].Available.https://www.ibm.com/analytics/spss-statistics-software
34. https://www.mathworks.com/

ԳԵՆԵՐԱՏԻՎ-ՄՐՑԱԿՑԱՅԻՆ ՑԱՆՑԵՐԻ ԿԻՐԱՌՈՒՄԸ ՍՏՈՒԱՍՏԻԿ ՏԵՂԱՓՈՒՎՈՂ ՃՇԳՐԻՏ ԺԱՄԱՆԱԿԻ ԱԶԴԱՆՇԱՆՆԵՐԻ ԱՂԲՅՈՒՐՆԵՐԻ ՂԵԿԱՎԱՐՄԱՆ ՀԱՄԱՐ

Թ.Վ. Ջամղարյան
Հայաստանի ազգային պոլիտեխնիկական համալսարան

Ներկայացված են ապակենտրոնացված ցանցային ենթակառուցվածքում ստոխաստիկ տեղափոխվող ճշգրիտ ժամանակի աղբյուրների կիրառության հնարավորության հաշվարկները և թեստերի արդյունքները:

Ճշգրիտ ժամանակի աղբյուրների կլիենտների տեղափոխման ղեկավարումը և միացումը ճշգրիտ ժամանակի սերվերներին իրականացվել է գեներատիվ-մրցակցային ցանցի հիման վրա: Իրականացվել է ճշգրիտ ժամանակի ազդանշանների աղբյուրների դեմ հասանելիության գրոհի մոդելավորում՝ իրենց ստոխաստիկ տեղափոխման ժամանակ:

Անցկացված տեսական (գծային հետընթածի հիման վրա) և գործնական (վիրտուալ միջավայրում) թեստերը ապացուցում են համակարգի օգտագործման հնարավորությունը բարձր դինամիկայով, ապակենտրոնացված ցանցային ենթակառուցվածքներում հասանելիության դեմ գրոհի զանգվածային կիրառման դեպքում: Մշակվել է հաշվեկարգ և համապատասխան ծրագրային ապահովում գեներատիվ-մրցակցային ցանցի գործունեության ապահովման համար:

Բանալի բառեր. հետընթած, գեներատիվ-մրցակցային ցանց, հասանելիության դեմ գրոհ, ճշգրիտ ժամանակի ազդանշան:

ПРИМЕНЕНИЕ ГЕНЕРАТИВНО-СОСТЯЗАТЕЛЬНЫХ СЕТЕЙ ДЛЯ УПРАВЛЕНИЯ СТОХАСТИЧЕСКИ ПЕРЕМЕЩАЕМЫМИ ИСТОЧНИКАМИ СИГНАЛО ТОЧНОГО ВРЕМЕНИ

Т.В. Джамгарян

Национальный политехнический университет Армении

Представлены результаты расчетов и тестов применения стохастически перемещаемых источников сигналов точного времени в децентрализованной сетевой инфраструктуре. Управление перемещением и подключением клиентов точного времени к серверам точного времени осуществлялось на основе генеративно-состязательной сети.

Проведено моделирование атаки на доступность на источники сигналов точного времени при их стохастическом перемещении. Проведенные теоретические (на основе линейной регрессии) и практические тесты (в виртуальной среде), доказывают возможность применения системы в высокодинамичных децентрализованных сетевых инфраструктурах при массированных атаках на доступность.

Разработан алгоритм и соответствующее програмнное обеспечение для функционирования управляющей генеративно-состязательной сети.

Ключевые слова: регрессия, генеративно-состязательная сеть, атака на доступность, сигнал точного времени.

Submitted on 17.08.2021.
Sent for review on 19.08.2021.
Guaranteed for printing on 07.10.2021.