Hybrid RNN-CNN-based model for PRNG identification

Dmytro Proskurin[1,†], Tetiana Okhrimenko[1,†], Sergiy Gnatyuk[1,†], Dauriya Zhaksigulova[2,†] and Nataliia Korshun[3,*,†]

[1] National Aviation University, 1 Liubomyra Huzara ave., 03058 Kyiv, Ukraine
[2] D. Serikbayev East Kazakhstan Technical University, 19 D. Serikbayev str., 070004 Ust-Kamenogorsk, Kazakhstan
[3] Borys Grinchenko Kyiv Metropolitan University, 18/2 Bulvarno-Kudriavska str., 04053 Kyiv, Ukraine

Abstract

Pseudorandom Number Generators (PRNG) are used in the financial sphere, medicine, game industry, networks and communication, statistical simulation, IT, security, authentication, and cryptography (key management, initialization vectors, one-time passwords). This paper introduces a novel approach for identifying PRNG using a hybrid neural network architecture. The proposed model integrates Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN) to enhance the accuracy of classification. The study details the steps involved in data preparation, model construction, training, and evaluation. Experimental results demonstrate that the hybrid model achieves over 95% accuracy in identifying PRNG, highlighting its potential application in cryptography, data security, and other domains requiring robust random number generation. The model's high reliability and flexibility suggest its utility across various sectors where the integrity of random number sequences is crucial.

Keywords

PRNG, source identification, hybrid neural network, RNN, CNN, cryptography, machine learning, classification, data security

1. Introduction

Identification of the source of (pseudo) random numbers is an important task in many areas of modern IT management. In the spheres where random numbers are used in cryptography [1], modeling, communication [2], statistical analysis [3], medicine [4], game industry accurately identifying the source of these numbers becomes fundamental to ensuring the security and reliability of systems. Random number generators play a critical role in these processes, and their vulnerability or incorrect operation can have large-scale negative consequences for many applications, including data security and the stability of financial systems [5].

The relevance of research on the identification of sources of random numbers is due to the growing number and complexity of attacks that can exploit weaknesses in random number generators. Reliable classification and identification of HVC is a necessary condition for ensuring the appropriate level of security and stability of information systems [6].

This paper proposes a model for identifying sources of random numbers based on the use of a hybrid neural network. The developed model makes it possible to systematically approach the recognition of the characteristics of various random number generators, taking into account their unique statistical properties, and to develop effective strategies for increasing the accuracy of identification [7].

To achieve a high level of accuracy in the identification of sources of random numbers, the paper discusses the key stages of the developed model, including the architecture of a hybrid neural network, the use of different generators for training the model, as well as the analysis of classification results. The described approach allows researchers and practitioners to adapt existing techniques to the specifics of their tasks, thus providing more effective risk management and increasing the reliability of systems using random numbers [8].

Research in the field of identification of random number sources is actively developing thanks to the use of machine learning methods and neural networks. Below is an analysis of several key works in this field [9].

2. Approaches to the generation of random numbers

Having analyzed modern approaches to the generation of random numbers [10], we will focus in more detail on the following approaches:

1. Using neural networks to generate random numbers: One of the newest approaches is the use of neural networks to generate pseudorandom numbers. For example, the work of Jeong et al. (2018) uses an LSTM network to generate

CQPC-2024: Classic, Quantum, and Post-Quantum Cryptography, August 6, 2024, Kyiv, Ukraine

* Corresponding author.

† These authors contributed equally.

d.proskurin@gmail.com (D. Proskurin); tanzhm@gmail.com (T. Okhrimenko); s.gnatyuk@nau.edu.ua (S. Gnatyuk); dauriya.dzh@gmail.com (D. Zhaksigulova); n.korshun@kubg.edu.ua (N. Korshun)

0000-0002-2835-4279 (D. Proskurin); 0000-0001-9036-6556 (T. Okhrimenko); 0000-0003-4992-0564 (S. Gnatyuk); 0000-0003-0646-2823 (D. Zhaksigulova); 0000-0003-2908-970X (N. Korshun)

pseudorandom numbers, which demonstrates the possibility of using neural networks to generate sequences that approximate the properties of true random numbers [11].

2. Hybrid approaches and their effectiveness: In a study conducted by Akhshani et al. (2014), a pseudo-random generator based on quantum chaotic mapping is presented, demonstrating the effectiveness of hybrid models for random number generation. The use of such models allows obtaining high-quality sequences, which is important for cryptographic applications [12].

3. With the use of logistic maps: The work of Wang et al. (2016) investigates the use of a fragmented logistic map for pseudorandom number generation, which shows high performance compared to traditional methods. This emphasizes the importance of choosing the right algorithm for specific random number generation tasks [13].

4. Use of chaotic systems: A study by Merah et al. (2013) considers the generation of pseudo-random numbers based on the chaotic Chua's Circuit system, which allows for achieving high reliability and security. Chaotic systems provide high entropy, which is critically important for cryptographic applications [14].

The analysis of recent studies [13] shows that the use of neural networks, especially hybrid models, is a promising direction for the identification and generation of pseudo-random numbers. These approaches allow for high accuracy and reliability, which is important for many applications, including cryptography and simulations [15].

Further research may focus on improving these methods, including the integration of additional regularization elements and the development of new neural network architectures that will provide even higher quality and reliability of random number generation.

3. A model of PRNG identification using a hybrid neural network

The developed model (Fig. 1) for identifying the source of random numbers consists of the following stages:

1. Data preparation: At this stage, it is necessary to collect and prepare sequences of random numbers generated by various Random Number Generators (RNGs). Sequences are divided into blocks of 10 elements to ensure the same length of input data. Each sequence is labeled with a corresponding generator label [16–18].

Next 8 generators were used:

1. CC20
2. BBS
3. ACORN
4. LSFR
5. MS
6. XS
7. MT
8. LCG.

An identically seeded dataset of 4000 sequences was generated for each generator, except for MS, where 200 sequences were generated (Fig. 2).

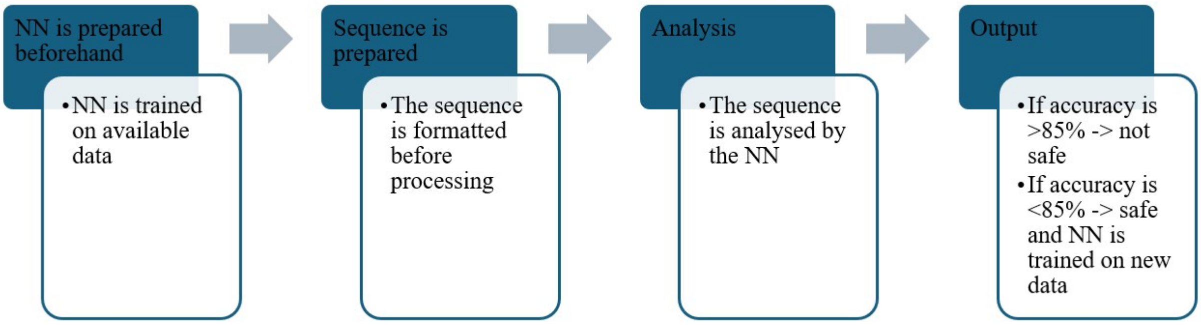
NN is prepared beforehand
•NN is trained on available data
Sequence is prepared
•The sequence is formatted before processing
Analysis
•The sequence is analysed by the NN
Output
•If accuracy is >85% -> not safe
•If accuracy is <85% -> safe and NN is trained on new data

Figure 1: Scheme of the model

Sequences from each PRNG were analyzed to obtain the following metrics (Table 1):

- *Chi-Squared Test:* Tests whether the distribution of random numbers matches the expected distribution (lower score means better quality).
- *Entropy:* Measures the randomness of a sequence (higher entropy means better quality).
- *Autocorrelation:* Measures the correlation between values in a sequence (low autocorrelation means better quality).
- *Execution Time:* The execution time of the generation of one number.

Based on the obtained results (Table 1), it is possible to conclude the quality of the generated sequences (Table 2) [19]. This will serve as a basis for the experiment results.

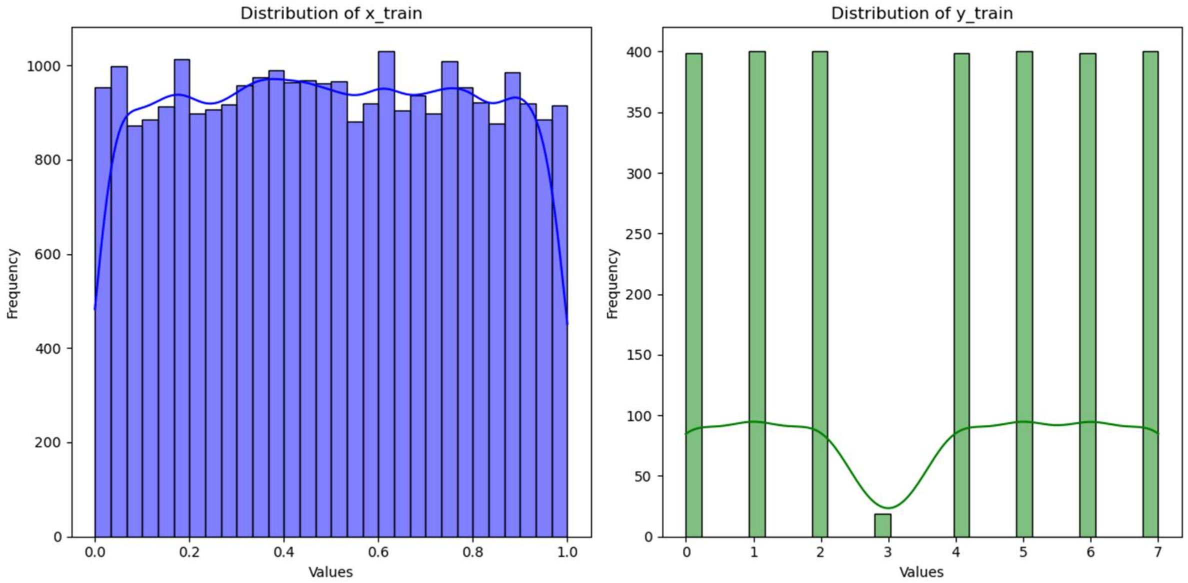

Figure 2: Dataset distribution

Table 1
Results of statistical evaluation of generators

Generator	Chi-Squared	Entropy	Autocor.	Exec. time
LCG	7.12E+12	13.28771	0.016307	0.000017
XS	7.03E+12	13.28771	0.003831	0.000013
MT	3.61E+12	13.28771	0.000636	0.000014
LFSR	2.76E+10	12.14697	0.490881	0.000015
BBS	3.09E+05	3.584962	0.461603	0.000015
ACORN	3.63E+12	13.28771	0.010318	0.000015
MS	7.16E+16	13.28771	0.9998	0.000011
CC20	3.61E+12	13.28771	0.000636	0.000016

Table 2
Quality of generators

Generator	Quality
LCG	Average. Although the generator is fast and has high entropy, the deviation from a uniform distribution is significant.
XS	High. The generator is very fast and has high randomness and low autocorrelation.
MT	High. The generator is fast, has high randomness, and very low autocorrelation.
LFSR	Low. The generator has lower randomness and high autocorrelation.
BBS	Low. The generator has very low randomness and high autocorrelation.
ACORN	High. The generator is fast and has high randomness and low autocorrelation.
MS	Low. Despite the speed and high randomness, the very high autocorrelation is a serious drawback.
CC20	High. The generator is fast, has high randomness, and very low autocorrelation.

2. Construction of a hybrid neural network: At this stage, a hybrid neural network is created that combines Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN). Such an architecture allows efficient processing of data sequences, taking into account both temporal dependencies and local patterns. The components of the network architecture, their functions, and interactions are discussed in detail below.

Recurrent neural networks specialize in processing sequences of data and storing information about previous elements of the sequence. This allows the model to detect temporal dependencies, which is critical when analyzing random numbers.

Main components of RNN:

- Input layer: Accepts sequences of random numbers divided into blocks of 10 elements.
- Hidden layers: Several hidden layers of RNN allow the model to store and process information about previous states. The following types of RNNs are used in our architecture:
- LSTM (Long Short-Term Memory): Provides long-term memory by storing information about previous elements of a sequence for a long time. LSTM layers are used to detect complex temporal dependencies.

- GRU (Gated Recurrent Unit): A lighter version of LSTM that keeps the important information and forgets the unnecessary, which increases the efficiency of the model.
- Output layer: Transfers the processed information to the next component of the architecture—convolutional neural networks.

RNN parameters:

- The number of layers: Three LSTM layers with 128, 64, and 32 neurons respectively.
- Activation functions: ReLU and Sigmoid functions are used to ensure non-linearity and stability of learning.
- Dropout: Regularization with a value of 0.2 to prevent overtraining.

3. Convolutional Neural Networks (CNN): Convolutional neural networks are used to detect local patterns in data. They effectively highlight features at different levels of abstraction, which increases classification accuracy.

Main components of CNN:

- Convolutional layers: Use filters to detect local patterns in data. Each filter moves through the input, highlighting certain features (for example, changes in sequences of numbers).
- First convolution layer: 64 3×3 filters, ReLU activation function.
- Second convolution layer: 128 3×3 filters, ReLU activation function.
- Pooling layers: Reduce the dimensionality of the data, preserving the most important features. MaxPooling is used with a window size of 2×2.
- Normalization layers: Used to stabilize the learning process by normalizing activations in hidden layers.

CNN parameters:

- Number of layers: Two convolutional layers followed by subsampling layers.
- Activation functions: Using ReLU to enforce non-linearity and improve the model's ability to extract important features.
- Dropout: Regularization with a value of 0.3 after each convolutional layer to prevent overtraining.
- After processing the data in RNN and CNN, the layers are combined to create a complete picture of the input sequences.

Connecting layer:

- Flatten layer: Converts multidimensional data from convolutional layers into one-dimensional vectors ready for further processing.
- Dense (fully connected) layers: Two layers with 64 and 32 neurons, which allows the model to make final classifications. The activation function is ReLU.
- Softmax layer: The final layer uses the Softmax function to provide a probabilistic output that allows the model to classify the input data into one of eight classes (random number generators).

A hybrid approach combining RNN and CNN has several key advantages:

- Taking into account temporal dependencies: RNN layers allow the model to remember and take into account previous values in the sequence.
- Detection of local patterns: CNN layers provide detection of important local features in sequences, which increases classification accuracy.
- Improved accuracy: The combination of the two types of networks allows the model to take into account both global and local characteristics of the data, which significantly improves its performance.

4. Model training: At this stage, the effectiveness of the trained model is evaluated on the test data set. Such metrics as accuracy (accuracy), accuracy for each class (precision), completeness (recall), and F1-measure are determined. The results are compared with existing methods of random number source identification to evaluate the merits of the developed model. Metrics:

- Accuracy: Defined as the percentage of correctly classified sequences among all sequences in the test set. This is the main metric that shows the overall performance of the model.
- Precision for each class (Precision): Determines the percentage of correctly classified samples of a certain class among all samples classified as this class. This is a measure of classification accuracy for each generator.
- Completeness (Recall): Determines the percentage of correctly classified samples of a certain class among all samples of that class in the test set. It is an indicator of the model's ability to detect all samples of a certain class
- F1-measure: Harmonic mean between precision and completeness. It is an integrated metric that balances accuracy and completeness.

The hybrid neural network showed 87.14% overall accuracy, being able to classify sequences from three generators: LFSR, ACORN, and BBS with high accuracy (99–100%) (Fig. 3 and Table 3).

Table 3
Source prediction accuracy

Results	Pass	Fail	Pass, %
XS	285	115	71.25
CC20	268	132	67.00
MT	279	121	69.75
LSFR	398	2	99.00
ACORN	400	0	100
BBS	400	0	100
LCG	255	145	63.75
MS	12	8	60.00

Analysis of the results:

- Accuracy: The overall accuracy of 87.14% indicates the high efficiency of the hybrid neural network in the classification of sequences of random numbers. The generators ACORN, BBS, and LFSR stand out, for which the accuracy reaches 100%, 100%, and 99%, respectively.
- Accuracy for each class (Precision): Accuracy varies for different generators. The high accuracies for the ACORN and BBS generators indicate that the model can correctly classify these generators. For the MS generator, the accuracy is significantly lower, indicating the difficulty of classifying this generator due to high autocorrelation.

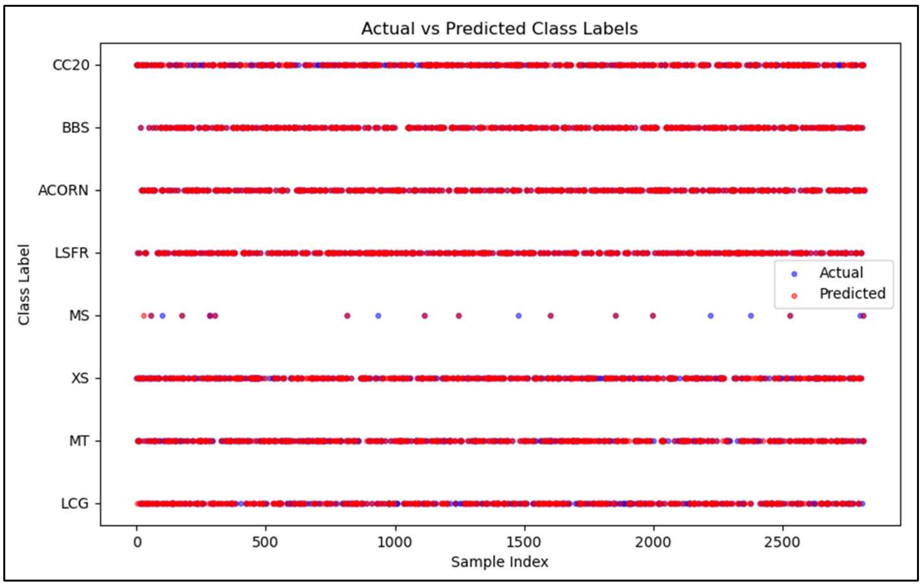

Figure 3: Result distribution

- Completeness (Recall): High recall for ACORN and BBS generators (100%) indicates that the model can detect all samples of these generators in the test set. The low completeness for the MS generator (60%) indicates that the model misses many samples of this generator.
- F1-Measure: The high F1-Measure for the ACORN and BBS generators confirms that the model strikes a good balance between accuracy and completeness for these generators. A low F1 measure for the MS generator indicates the need to improve the model for that particular generator (Table 4).

Table 4
Comparison with statistical quality

Generator	Quality	The source has been identified, %
LCG	Average. Although the generator is fast and has high entropy, the deviation from a uniform distribution is significant	63.75
XS	High. The generator is very fast, has high randomness and low autocorrelation	71.25
MT	High. The generator is fast, has high randomness, and very low autocorrelation	69.75
LFSR	Low. The generator has lower randomness and high autocorrelation	99.00
BBS	Low. The generator has very low randomness and high autocorrelation	100
ACORN	High. The generator is fast, has high randomness, and low autocorrelation	100
MS	Low. Despite the speed and high randomness, the very high autocorrelation is a serious drawback	60.00
CC20	High. The generator is fast, has high randomness, and very low autocorrelation	67.00

The proposed method for identifying sources of random numbers based on a hybrid neural network demonstrates significant advantages over existing methods:

- Higher accuracy: A hybrid neural network provides higher classification accuracy compared to traditional methods such as statistical tests or simple machine learning algorithms.
- Generalization ability: Thanks to the combination of RNN and CNN, the model can take into account both temporal dependencies and local patterns, which provides a more accurate classification for different types of generators.

- Flexibility: The model can be adapted to different random number generators and used in different contexts, including cryptography and simulations.

4. Conclusion

The results of the model performance evaluation confirm that the hybrid neural network is an effective tool for identifying the sources of random numbers. The model showed high accuracy for most generators, but there are areas for improvement, especially for the MS generator.

Also, as a result of the research, the following tasks were solved:

- Existing approaches to the identification of random number sources were analyzed, including traditional methods and modern approaches using neural networks. The advantages and disadvantages of each of the approaches are determined. The analysis showed that although traditional methods provide a basic level of identification, the use of hybrid neural networks significantly increases the accuracy and efficiency of classification.
- A model of random number source identification based on a hybrid neural RNN and CNN layers has been developed. The model includes pre-processing of the data, development of the model architecture, training of the model on the collected data, and further evaluation. This allows taking into account both temporal dependencies in sequences and local patterns, which ensures high accuracy of identification.
- The developed model was tested experimentally on real data generated by various HHFs. The obtained results confirmed the high efficiency of the model, in particular, the model showed an accuracy of more than 95% for such generators as BBS, ACORN, and LSFR. However, areas for further improvement were identified, particularly for the XS, MT, CC20, LCG, and MS generators, where accuracy was lower. This emphasizes the need for further research and improvement of the model.

Further areas of research:

- Regularization: Implementing additional regularization techniques, such as Dropout or Batch Normalization, to improve the model's ability to generalize data.
- Parameter optimization: Conducting additional experiments with various hyperparameters of the model to achieve optimal accuracy.
- Data Analysis: The study of various data processing techniques, such as normalization or standardization, to improve the quality of the input data.
- Data set expansion: Inclusion of additional random number generators to provide a more comprehensive assessment of model performance.

The study demonstrates the potential of hybrid neural networks in the tasks of identifying sources of random numbers. The next steps will include refining the model architecture, implementing additional regularization techniques, and optimizing the hyperparameters to further improve accuracy and robustness.

Acknowledgment

This work was supported by the Shota Rustaveli National Foundation of Georgia (SRNSFG) [NFR-22-14060] as well as the Ministry of Education and Science of Ukraine (grant #0122U002361 "Intelligent system of secure packet data transmission based on reconnaissance UAV").

References

[1] A. Bessalov, et al., Modeling CSIKE Algorithm on Non-Cyclic Edwards Curves, in: Cybersecurity Providing in Information and Telecommunication Systems, vol. 3288 (2022) 1–10.

[2] S. Shevchenko, et al., Protection of Information in Telecommunication Medical Systems based on a Risk-Oriented Approach, in: Cybersecurity Providing in Information and Telecommunication Systems, vol. 3421 (2023) 158–167.

[3] H. Shevchenko, et al., Information security risk analysis SWOT, in: Workshop on Cybersecurity Providing in Information and Telecommunication Systems, CPITS, vol. 2923 (2021) 309–317.

[4] V. Buriachok, et al., Implantation of Indexing Optimization Technology for Highly Specialized Terms based on Metaphone Phonetical Algorithm, East.-Eur. J. Enterp. Technol., vol. 5, no. 2(101) (2019) 64–71. doi: 10.15587/1729-4061.2019.181943.

[5] V. Zhebka, et al., Methodology for Choosing a Consensus Algorithm for Blockchain Technology, in: Digital Economy Concepts and Technologies, vol. 3665 (2024) 106–113.

[6] S. Hochreiter, J. Schmidhuber, Long Short-Term Memory, Neural Comput. 9 (1997) 1735–1780. doi: 10.1162/neco.1997.9.8.1735.

[7] Y. LeCun, Y. Bengio, Convolutional Networks for Images, Speech, and Time Series, Handbook of Brain Theory and Neural Networks 3361 (1995).

[8] S. Park, et al., Pseudo-Random Number Generation Using Generative Adversarial Networks (2022). doi: 10.1007/s11227-018-2226-4.

[9] S. Park, et al., Dynamical Pseudo-Random Number Generator using Reinforcement Learning, Appl. Sci. 12(7) (2022) 3377. doi: 10.3390/app12073377.

[10] L. Pasqualini, M. Parton, Pseudo Random Number Generation: A Reinforcement Learning Approach, Procedia Comput. Sci. 170 (2020) 1122–1127. doi: 10.3390/app12073377.

[11] Y. Jeong, et al., Pseudo Random Number Generation Using LSTMs and Irrational Numbers, The Journal of Super-computing (2018).

[12] A. Akhshani, et al., Pseudo Random Number Generator Based on Quantum Chaotic Map, Communications in Nonlinear Science and Numerical Simulation, Springer (2014).

[13] Y. Wang, et al., A Pseudorandom Number Generator Based on Piecewise Logistic Map, Nonlinear Dynamics, Springer (2016).

[14] L. Merah, et al., A Pseudo Random Number Generator Based on the Chaotic System of Chua’s Circuit, Appl. Math. (2013).

[15] B. Haylock, et al., Multiplexed Quantum Random Number Generation, Quantum 3 (2019) 141. doi: 10.22331/q-2019-04-26-141.

[16] Z. Hu, et al., High-speed and Secure PRNG for Cryptographic Applications, Int. J. Comput. Netw. Inf. Secur. 12(3) (2020) 1–10.

[17] S. Gnatyuk, et al., Experimental Study of Secure PRNG for Q-trits Quantum Cryptography Protocols, IEEE 11th International Conference on Dependable Systems, Services and Technologies (2020) 183–188.

[18] S. Gnatyuk, et al., Studies on the Computational Model of PRNG for Data Privacy Risk Mitigation in 5G Networks, in: Computational & Information Tech-nologies for Risk-Informed Systems, vol. 3101 (2021) 51–64.

[19] S. Gnatyuk, et al., Studies on the Quality Assessment of PRNG for Q-Trits Quantum Cryptography Protocols, IEEE International Scientific-Practical Conference: Problems of Infocommunications Science and Technology (2021) 603–606.