Scientific and Practical Cyber Security Journal (SPCSJ) 8(4): 110 – 117 ISSN 2587-4667
Scientific Cyber Security Association (SCSA)

ML-BASED PREDICTING PRNG OUTPUT WITH SEQUENTIAL ANALYSIS

Dmytro Proskurin[1], Maksim Iavich[2], Sergiy Gnatyuk[1], Tetiana Okhrimenko[1],
[1]State University "Kyiv Aviation Institute", [2]Caucasus University

ABSTRACT: This research goes into the predictive capabilities of neural network models, mainly focusing on recurrent neural networks (RNNs) and long-term short-term memory networks (LSTMs), and their combination in a hybrid architecture to predict the outcomes of various pseudo-random number generators (PRNGs). In this work Continuous-Output Scenario Analysis is shown.

KEYWORDS*: Random numbers, Neural Networks, AI, Hybrid model, PRNG, cybersecurity, critical information infrastructure.*

INTRODUCTION

Neural networks are artificial intelligence systems designed to replicate the functioning of the human brain. Unlike traditional digital models that process binary data (zeros and ones), neural networks operate through connection processing units, akin to neurons in a biological brain. The performance of these networks depends on how their connections are structured and weighted. Neural networks are algorithms modelled after the human brain that recognize patterns. Sensory data is interpreted using machine perception, which labels or clusters raw information. They recognize numerical patterns in vectors, which must be converted into real-world data like as images, sounds, text, and time series [1]. *Artificial Neural Networks (ANNs)* are computing systems modelled such as biological neural systems, including the human brain. These systems consist of numerous interconnected computational units (neurons) that work collaboratively in a distributed manner. By learning from input data and optimizing outputs, they enhance performance and achieve accurate results [2].

Let's consider the main Neural Networks types:

1) *Convolutional Neural Networks (CNNs)* are comparable to standard ANNs because they utilize neurons and improve through learning [2]. CNNs have achieved significant breakthroughs and are now a key component of deep learning applications. They have revolutionized computer vision, enabling advancements such as facial recognition, autonomous driving, cashier-less retail systems, and intelligent healthcare technologies. Unlike traditional ANNs, CNNs are tailored for recognizing patterns in visual data. This specialization allows them to embed image-specific features directly into their structure, making them particularly effective for image-centric tasks. Furthermore, CNNs require fewer parameters to configure, enhancing their efficiency and performance in complex visual processing tasks [2, 3].
2) *Hybrid neural networks (HNNs)* are becoming increasingly popular in computer vision applications including picture captioning and action identification, and they integrate the strengths of many neural networks. However, there has been limited research on the effective use of hybrid architectures for time series data, particularly for trend forecasting purposes [4]. HNNs use their internal structure to limit the interactions between process variables in order to align with physical models. Compared to regular neural networks, coupled models are more accurate, dependable, and generalizable [5].
3) *Recurrent Neural Networks* (RNNs) represent a paradigm shift in neural networks, specifically designed to recognize patterns in sequences of data [6]. Unlike traditional feedforward neural networks, RNNs possess a unique feature: the output from the previous step is fed back into the output of the current step. This looping mechanism allows RNNs to maintain an internal state that captures information about the sequence they have processed so far, making them ideal for tasks like speech recognition, language modelling, and time series forecasting [6 7]. The core architecture of an RNN involves a hidden layer where the activation at a given time step is a function of the output at the same step and the activation of the hidden layer at the previous step [8]. This recurrent nature allows the network to maintain a form of memory [6]. However,

RNNs are often challenged by long-term dependencies due to issues like vanishing and exploding gradients during backpropagation [7, 9], where the network becomes unable to learn and retain information from earlier time steps in the sequence [10].

4) *Long Short-Term Memory networks*, a special kind of RNN, were developed to overcome the limitations of traditional RNNs. LSTMs are adept at learning long-term dependencies, thanks to their unique internal structure [9]. Unlike standard RNNs, LSTMs have a complex architecture with a series of gates: the forget gate, output gate, and output gate [9, 10]. These gates regulate the flow of information into and out of the cell, deciding what to keep in memory and what to discard, thereby addressing the vanishing gradient problem [8].

Both RNNs and LSTMs are designed for sequence processing, the key difference lies in their ability to handle long-term dependencies. Standard RNNs, while simpler and computationally less intensive, struggle with retaining information over longer sequences. LSTMs, with their intricate gating mechanism, excel in scenarios where understanding long-range contextual information is crucial.

The choice between RNNs and LSTMs often boils down to the specific requirements of the task at hand, the complexity of the sequences involved, and the computational resources available. LSTMs are generally preferred for more complex tasks with longer sequences [9], while RNNs might suffice for simpler tasks with shorter temporal dependencies [11].

In our research, we utilized datasets produced by four different PRNG algorithms, each presenting unique challenges and features for sequence prediction with RNN and LSTM models. These datasets provided a platform to assess and contrast the performance of various neural network architectures in tackling sequence prediction tasks.

After generating the dataset, it was carefully divided into three distinct subsets to support the training, testing, and validation of our predictive models:

- **Training Set**: Used for model training, enabling the networks to learn and adapt to the patterns present in the pseudorandom sequences generated by each PRNG.
- **Testing Set**: Reserved for evaluating the models' performance on unseen data, providing an objective measure of their predictive accuracy.
- **Validation Set**: Applied during the model tuning process to adjust parameters and mitigate overfitting, ensuring the models can generalize effectively to new data.

This structured approach to dataset preparation and partitioning played a pivotal role in establishing a reliable framework for analyzing the predictability of PRNG outputs via sequential analysis. By standardizing generation parameters and implementing a methodical data split, we ensured a consistent and equitable evaluation environment for all predictive models used in the study.

EVALUATION METRICS FOR TESTING

To evaluate how effectively our models predict PRNG outputs, we utilized a range of detailed assessment metrics. These metrics play a vital role in measuring prediction accuracy and enabling direct comparisons between the various neural network architectures examined in our research. The core of our evaluation framework includes the Mean Squared Error (MSE) and a custom-designed Model Performance Score.

MSE is a fundamental component of our evaluation approach, providing a robust measure of prediction accuracy. It calculates the average squared differences between predicted and actual values, highlighting the magnitude of prediction errors. By emphasizing larger discrepancies through squaring, MSE becomes particularly sensitive to outliers and significant inaccuracies.

When applied to predicting PRNG outputs, MSE offers a straightforward metric to assess how closely the model's predictions match the actual sequence of numbers produced by the PRNGs. A lower MSE signifies greater prediction accuracy, indicating the model's effectiveness in capturing and replicating the underlying patterns of the PRNG sequence [13-15].

Recognizing the need for a standardized metric that allows for an intuitive understanding of model performance, we introduced the Model Performance Score. This metric normalizes the MSE to a scale ranging from 0 to 1, where 0 represents the poorest performance (highest MSE) and 1 denotes perfect prediction accuracy (zero MSE).

The Model Performance Score is calculated by inversely scaling the MSE against a predetermined maximum error threshold. This approach ensures that the performance score is adjusted for the scale of

the data and the expected variation in prediction accuracy, allowing for a fair comparison across different models and datasets.

This normalized score simplifies the interpretation of our results, providing a straightforward metric to gauge model effectiveness. It allows stakeholders to quickly assess the relative performance of each model in predicting PRNG outputs without delving into the complexities of raw MSE values.

Together, these evaluation metrics form the foundation of our analytical approach, enabling a nuanced analysis of model performance. MSE offers a detailed view of the prediction accuracy, while the Model Performance Score provides a high-level, comparative perspective. By incorporating both metrics, our study ensures a balanced and comprehensive evaluation of how well each neural network architecture can predict the seemingly unpredictable: the output of pseudorandom number generators.

We carried out a comprehensive set of experiments to assess the predictive performance of various neural network architectures. These experiments were carefully structured to examine how different model parameters influence the accuracy of PRNG output predictions. In the following sections, we outline the key variables considered in these experiments and discuss significant insights regarding model performance.

To systematically assess the effects of various hyperparameters on model performance, we tested a wide array of combinations, encompassing:

- *Activation Functions*: experimented with two popular activation functions, ReLU (Rectified Linear Unit) and tanh (Hyperbolic Tangent). These functions were chosen for their distinct characteristics in handling nonlinearities in the data.
- *Number of Neurons*: the neuron counts tested were 8, 16, and 32. This range allowed us to explore the models' capacity to learn and generalize from the data, balancing complexity with computational efficiency.
- *Epochs:* all models were trained for [1000] epochs, providing ample opportunity for learning and convergence.
- *Model Layers:* varied the depth of the models by testing configurations with [2, 3, 5] layers. This variation aimed to understand how model depth influences learning and prediction accuracy.
- *Output Lengths:* for continuous value prediction, output lengths of [1, 2, 3, 5] were tested. This range was selected to assess the models' ability to forecast multiple steps ahead in the PRNG sequence.

EXPERIMENT

The study on predicting the output of PRNGs using sequential analysis gave convincing results, which were obtained by analyzing the most effective models for each PRNG. In the following, we consider meaningful results for scenarios with a continuous output for different PRNGs: Xorshift, MT (Mersenne Twister), LCG (Linear Congruential Generator), and MiddleSquare.

The continuous-output models demonstrated even higher predictive accuracy than single-output, with the Hybrid model configured for continuous predictions (Hybrid-C) achieving remarkable success.

For the MiddleSquare PRNG, the Hybrid-C model with tanh activation, 16 neurons, 3 layers, and an output length of 3 achieved a near-perfect mean score of 0.9955.

Table 1. *MiddleSquare, continuous output results*

Scenario	Model type	Neuron	Activation function	Epochs	Layers	Output length	Mean score
MiddleSquare	Hybrid-C	16	tanh	1000	3	3	0.995479
MiddleSquare	Hybrid-C	16	tanh	1000	2	2	0.992588
MiddleSquare	Hybrid-C	8	tanh	1000	5	2	0.990003
MiddleSquare	Hybrid-C	8	relu	1000	5	1	0.988989
MiddleSquare	Hybrid-C	32	relu	1000	5	3	0.988566
MiddleSquare	Hybrid-C	8	tanh	1000	3	1	0.988066
MiddleSquare	Hybrid-C	16	relu	1000	2	2	0.986359
MiddleSquare	Hybrid-C	16	tanh	1000	5	3	0.985923
MiddleSquare	Hybrid-C	16	tanh	1000	5	2	0.985797
MiddleSquare	Hybrid-C	8	tanh	1000	5	1	0.984813

Only 29% of the models managed to achieve a success rate exceeding 90% (Fig. 1).

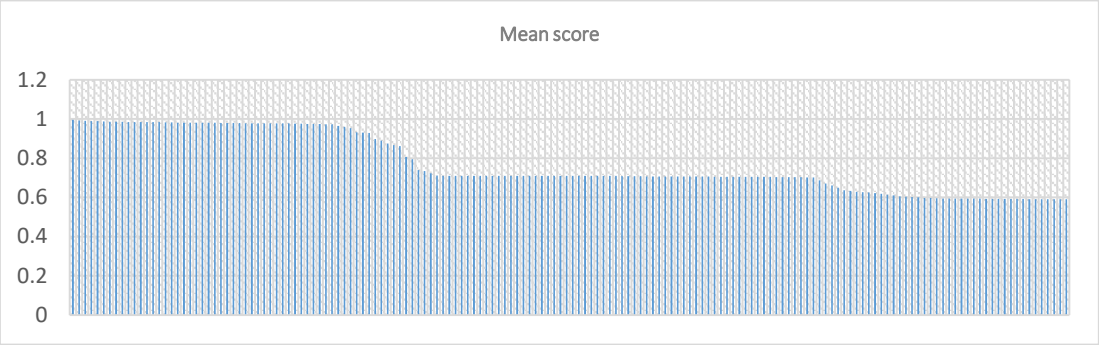

Fig. 11. *MiddleSquare, continuous output, all models results*

For the LCG PRNG, the Hybrid-C model with tanh activation, 8 neurons, 5 layers, and an output length of 2 achieved a near-perfect mean score of 0.992055.

Table 2. *LCG, continuous output results*

Scenario	Model type	Neuron	Activation function	Epochs	Layers	Output length	Mean score
LCG	Hybrid-C	8	tanh	1000	5	2	0.992055
LCG	Hybrid-C	8	tanh	1000	5	3	0.98973
LCG	Hybrid-C	16	tanh	1000	5	5	0.987614
LCG	Hybrid-C	8	tanh	1000	3	5	0.986818
LCG	Hybrid-C	8	tanh	1000	2	5	0.985174
LCG	Hybrid-C	16	tanh	1000	3	2	0.984808
LCG	Hybrid-C	32	tanh	1000	3	2	0.984462
LCG	Hybrid-C	16	tanh	1000	2	1	0.984411
LCG	Hybrid-C	32	tanh	1000	3	1	0.983866
LCG	Hybrid-C	32	tanh	1000	2	1	0.983706

20% of all models were able to break the 90% success threshold (Fig. 2).

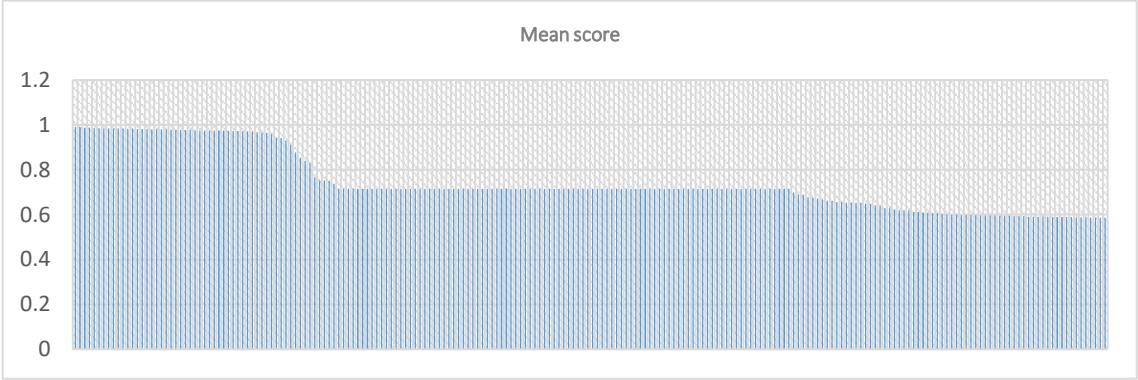

Fig. 2. *LCG, continuous output, all models results*

For the Xorshift PRNG, the Hybrid-C model with relu activation, 16 neurons, 2 layers, and an output length of 2 achieved a near-perfect mean score of 0.987906 (table 3).

Table 3. *Xorshift, continuous output results*

Scenario	Model type	Neuron	Activation function	Epochs	Layers	Output length	Mean score
Xorshift	Hybrid-C	16	relu	1000	2	2	0.987906
Xorshift	Hybrid-C	16	relu	1000	2	5	0.985753
Xorshift	Hybrid-C	8	relu	1000	5	5	0.985715
Xorshift	Hybrid-C	8	relu	1000	2	2	0.984238
Xorshift	Hybrid-C	32	relu	1000	2	2	0.983437
Xorshift	Hybrid-C	16	relu	1000	2	3	0.981247
Xorshift	Hybrid-C	8	relu	1000	2	1	0.980434
Xorshift	Hybrid-C	32	relu	1000	2	1	0.97893
Xorshift	RNN-C	32	tanh	1000	3	1	0.977685
Xorshift	Hybrid-C	32	relu	1000	2	3	0.976177

15% of all models were able to break the 90% success threshold (Fig. 3***Fig. 2***).

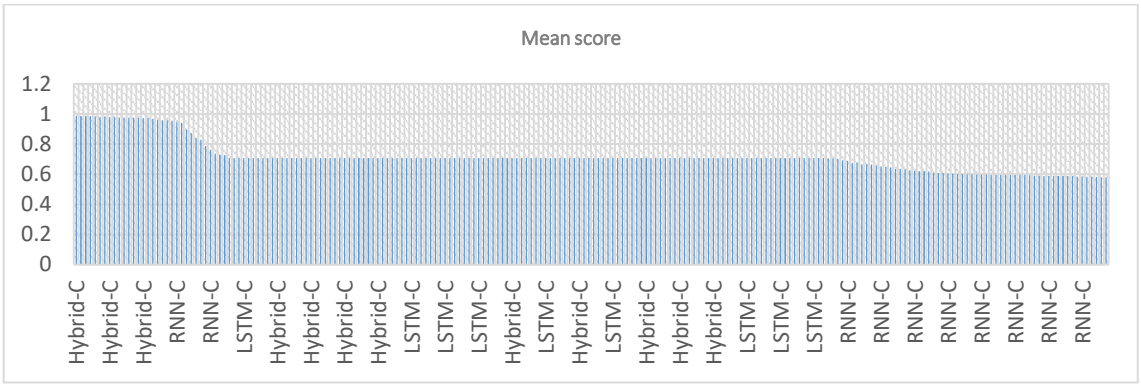

Fig. 2. *Xorshift, continuous output, all models results*

For the MT PRNG, the Hybrid-C model with relu activation, 32 neurons, 2 layers, and an output length of 2 achieved a near-perfect mean score of 0.985006 (table 4).

Table 4. *MT, continuous output results*

Scenario	Model type	Neuron	Activation function	Epochs	Layers	Output length	Mean score
MT	Hybrid-C	32	relu	1000	2	2	0.985006
MT	RNN-C	32	relu	1000	5	1	0.981523
MT	RNN-C	32	tanh	1000	5	1	0.980135
MT	Hybrid-C	16	tanh	1000	2	3	0.976324
MT	LSTM-C	32	relu	1000	5	1	0.976245
MT	Hybrid-C	8	tanh	1000	2	1	0.975258
MT	RNN-C	32	tanh	1000	3	1	0.97421
MT	RNN-C	32	relu	1000	3	1	0.972664
MT	RNN-C	32	relu	1000	2	1	0.965829
MT	RNN-C	32	tanh	1000	2	1	0.963914

12% of all models were able to break the 90% success threshold (Fig. 4***Fig. 2***).

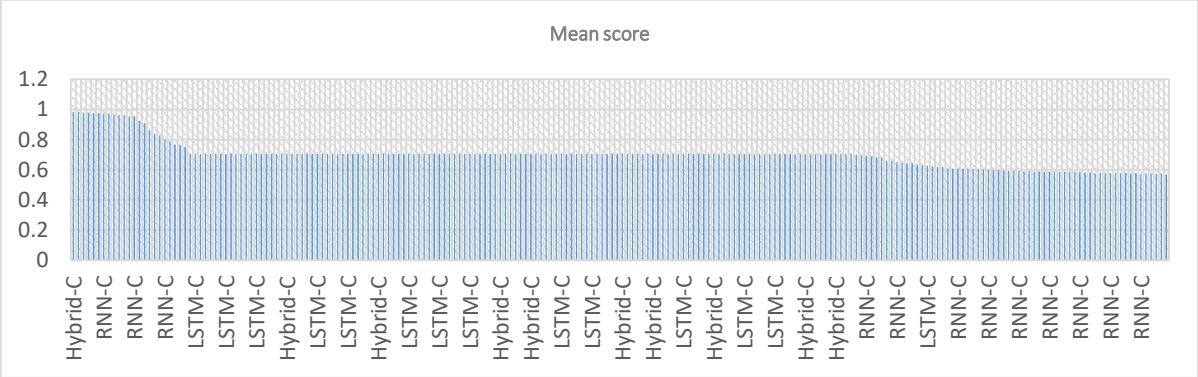

Fig. 3. *MT, continuous output, all models results*

The examination of continuous-output models reveals a notable enhancement in predictive performance compared to single-output models. This is particularly evident in the context of predicting sequences generated by the MiddleSquare PRNG.

The performance plot illustrating the correlation between predicted and actual values (Fig. ***4*** 5) for the continuous-output model shows an even tighter linear alignment than the single-output model. This near-perfect correlation, along with a high success score of 0.9955, reflects the model's exceptional predictive accuracy. The dense clustering of points along the diagonal suggests that the model can reliably predict the MiddleSquare PRNG's output with high confidence, and such precision is indicative of the model's ability to capture both the immediate and contextual dependencies within the PRNG's sequence.

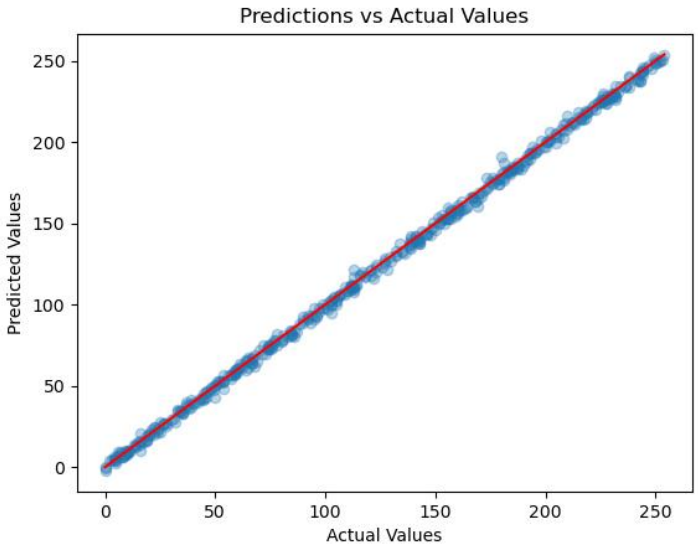

***Fig. 4**. Prediction vs actual values, MiddleSquare with the best result (0.9955)*

The scatter plot for the continuous-output Hybrid model, which integrates CNN and LSTM architectures (Hybrid-C), showcases a substantial concentration of points closely aligned with the line of perfect prediction (fig. 6). The model, employing tanh activation with 16 neurons across 3 layers, exhibits a remarkable ability to track the actual values throughout the sequence. This tight clustering indicates a substantial reduction in prediction errors and a strong alignment with the true PRNG sequence, suggesting a deeper understanding of the underlying patterns by the model.

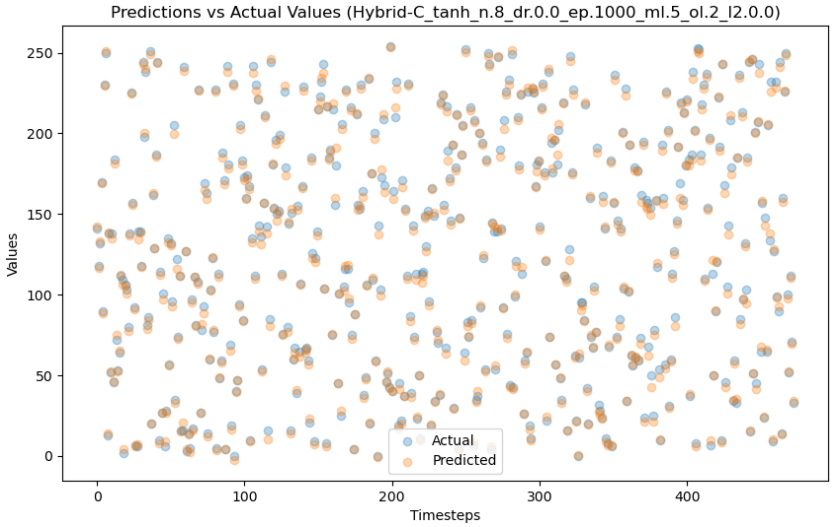

***Fig. 5**. Prediction vs actual values, MiddleSquare with the best result (0.9955)*

The continuous-output model's superior performance, as evidenced by the closer proximity of predicted to actual values and the higher success score, highlights the benefit of utilizing sequential context in PRNG output prediction. The ability to forecast the sequence with a success score reaching 0.9955 marks a significant milestone, suggesting that models incorporating sequence history can more effectively decode the deterministic yet complex structure of PRNG outputs.

This analysis implies that continuous-output models hold great promise for applications where forecasting accuracy over sequences is critical. The insights gleaned from this research can inform the development of more secure PRNGs, capable of withstanding sophisticated sequential analysis. Future work will likely explore the expansion of this approach to more complex and higher-dimensional

sequences, potentially integrating additional layers of complexity and exploring the impact on model performance.

Our study's findings highlight the nuanced nature of PRNG output prediction, with different models excelling for specific generators. This variation underscores the importance of model selection tailored to the characteristics of the PRNG being analyzed. For instance, the best-performing model for the Xorshift generator might leverage its unique XOR and shift operations, whereas the optimal model for the Mersenne Twister (MT) would need to account for its complex bit manipulation and tempering techniques.

Remarkably, the single-output models consistently achieved a 98% success rate across various PRNGs, demonstrating a high level of accuracy in predicting the next output value based solely on a single preceding value. This success rate is indicative of the models' ability to decipher the underlying deterministic patterns that govern PRNG outputs.

Even more impressive, the continuous-output model, which utilizes sequences of values to predict subsequent outputs, reached a 99% success rate. This improvement suggests that incorporating more context in the form of continuous output sequences enables the models to better capture the PRNGs' inherent algorithms, leading to more accurate predictions.

The success of our models in predicting PRNG outputs with such high accuracy has profound implications for the fields of cryptography and random number generation. While PRNGs are designed to produce sequences that are difficult to predict, our results suggest that advanced neural network models can uncover and exploit hidden patterns within these sequences. This finding calls for ongoing efforts to enhance the unpredictability and security of PRNGs, ensuring they remain robust against sophisticated analytical techniques.

CONCLUSION

This study investigates the predictability of PRNGs using advanced neural network architectures. Our analysis highlights the impressive ability of the tested models to accurately predict PRNG outputs, particularly in continuous output scenarios where their capacity to capture long-term dependencies within PRNG sequences proves highly effective. This underscores their potential for tackling complex sequence prediction challenges.

Future research should focus on integrating more sophisticated neural network architectures and exploring practical applications of these insights in areas such as secure communication and cryptographic key generation. The results of this work point to a critical need for the development of more secure and less predictable PRNG designs, strengthening defenses against adversarial predictions and enhancing the reliability of cryptographic systems.

ACKNOWLEDGMENT

This work was supported by Shota Rustaveli National Science Foundation of Georgia (SRNSFG) - CG-24-220

REFERENCES

[1]. Islam, M., Chen, G., & Jin, S. (2019). An Overview of Neural Network. American Journal of Neural Networks and Applications, 5(1), 7-11. doi:10.11648/j.ajnna.20190501.12.
[2]. O'Shea, K., & Nash, R. (2015). An Introduction to Convolutional Neural Networks.
[3]. Li, Z., Yang, W., Peng, S., & Liu, F. (2021). A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects.
[4]. Lin, T., Guo, T., & Aberer, K. Hybrid Neural Networks for Learning the Trend in Time Series.
[5]. Psichogios, D. C., & Ungar, L. H. A Hybrid Neural Network-First Principles Approach to Process Modeling.
[6]. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention Is All You Need. Advances in Neural Information Processing Systems, 30.
[7]. Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to Sequence Learning with Neural Networks. Advances in Neural Information Processing Systems, 27.

[8]. Karpathy, A. (2015). The Unreasonable Effectiveness of Recurrent Neural Networks. Andrej Karpathy blog.
[9]. Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), 1735-1780
[10]. Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to Forget: Continual Prediction with LSTM. Neural Computation, 12(10), 2451-2471.
[11]. Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. arXiv:1406.1078.
[12]. Brownlee, J. (2018). Deep Learning for Time Series Forecasting: Predict the Future with MLPs, CNNs and LSTMs in Python. Machine Learning Mastery.
[13]. Maksim Iavich, Tamari Kuchukhidze, Giorgi Iashvili, Sergiy Gnatyuk Hybrid quantum random number generator for cryptographic algorithms. RADIOELECTRONIC AND COMPUTER SYSTEMS, 4, 2021. DOI: https://doi.org/10.32620/reks.2021.4.09
[14]. Maksim Iavich, Tamari Kuchukhidze, Giorgi Iashvili, Sergiy Gnatyuk, Razvan Bocu," Novel Quantum Random Number Generator with the Improved Certification Method ", International Journal of Mathematical Sciences and Computing(IJMSC), Vol.7, No.3, pp. 41-53, 2021. DOI: 10.5815/ijmsc.2021.03.05
[15]. Maksim Iavich, Tamari Kuchukhidze, Giorgi Iashvili, Sergiy Gnatyuk, Razvan Bocu," Novel Quantum Random Number Generator with the Improved Certification Method ", International Journal of Mathematical Sciences and Computing (IJMSC), Vol.7, No.3, pp. 41-53, 2021. DOI: 10.5815/ijmsc.2021.03.05