AMERICAN JOURNALS
Open Access, Peer Reviewed Journals

American Journal of Business Management, Economics and Banking
ISSN (E): 2832-8078 Volume 47, April - 2026

DESIGN AND DEVELOPMENT OF A SIMULATION MODEL FOR BANK NETWORK TRAFFIC FILTERING SYSTEMS BASED ON DPI TECHNOLOGY USING THE ANYLOGIC SOFTWARE

Giyasova Nargiza Botirovna
Chief Specialist, Department of Scientific Research,
Innovations and Training of Scientific and Pedagogical Personnel,
Tashkent State University of Economics

ABSTRACT

This article analyzes the characteristics of artificial neural network algorithms and neural network models. Single-layer neural networks are also considered. Although a single neuron is incapable of performing even simple perception processes, the computational power of neural networks emerges when multiple neurons are interconnected within a network. The processes of handling incoming network load using Hamming and Hopfield models are examined. The results of both simulation models demonstrate that network traffic filtering is essential, firstly, for resource optimization, secondly, for protecting service systems from threats and attacks, and thirdly, for reducing request processing time. As a result, both the speed and quality of service are significantly improved.

KEYWORDS

Single-layer neuron, bank network simulator, types of simulators, Hamming and Hopfield, DPI.

Introduction

Simulation modeling consists of two main stages: modeling for decision-making and analyzing the results obtained through modeling. This study analyzes the characteristics of neural network algorithms as well as neural network models. In particular, the capabilities of Hamming and Hopfield models in processing incoming network load are examined. As a result, an effective model—namely, the Hamming model—was selected, and DPI systems were designed based on this model. The results of the developed system indicate that the selected model performs efficiently, achieving network analysis accuracy of up to 99%. These findings suggest that applying neural networks within DPI systems significantly improves network performance and enhances overall reliability.

As observed, developing a real simulation model requires substantial effort. First, the model developer must determine the tasks to be solved using the model; modeling must always begin with clearly defining its purpose. The objective determines which processes of the real system should be modeled and represented, which should be abstracted, which characteristics must be considered, and how relationships between variables and model parameters should be defined. These aspects must be reflected in the model.

This stage can be characterized by the development of a conceptual (content) model, which includes structuring the model, identifying subsystems, defining initial components, and establishing relationships at each hierarchical level.

In simulation modeling, the structure of the model reflects the structure of the real system and represents the actual relationships between its components. The elements of the system, parameters, variables, their interrelations, and the laws governing their changes must be represented within the modeling environment.

If necessary, an animation of the modeled processes should be developed to better understand system behavior.

Next, the constructed model must be verified in terms of implementation correctness. The following step involves calibration or adjustment of the model, including data collection and measurement of system characteristics in the real environment. These data are then incorporated into the model in the form of parameters and their distributions.

Subsequently, the model must be validated by testing it under several scenarios where the characteristics of the real system are known or observable. The final stage of working with the model is conducting computer experiments, which is essentially the purpose of model creation.

In its simplest form, experimentation involves observing system behavior under different parameter values and recording the results. This type of modeling is often referred to as "what-if" or predictive analysis. Computer simulation not only allows forecasting but also helps identify which control actions lead to desirable outcomes. More complex experiments include sensitivity analysis, risk assessment of different management decisions, and optimization to determine the best parameters and conditions for system operation.

Single-layer Neural Networks

Single-layer neural networks are among the simplest neural models. Although a single neuron cannot perform even basic perception processes, the computational power emerges when multiple neurons are combined into a network.

It should be noted that the input nodes (represented as circles) only distribute incoming signals and do not perform computations; therefore, they are not considered a layer. The neurons that perform computations are typically represented as rectangular units.

Each element of the input vector X is connected to each neuron through an individual weight. Each neuron computes a weighted sum of its inputs.

It is convenient to represent weights as elements of a matrix W, where the matrix has mmm rows and n columns (mmm is the number of inputs, n is the number of neurons). For example, $w_{i,j}$ represents the weight connecting the input to the neuron.

Thus, the output vector N, representing neuron outputs, can be calculated as a matrix product: $N=XW$ where N and X are row vectors. In solving problems using single-layer neural networks, the "winner-takes-all" principle is widely applied, where the neuron with the highest activation determines the final output.

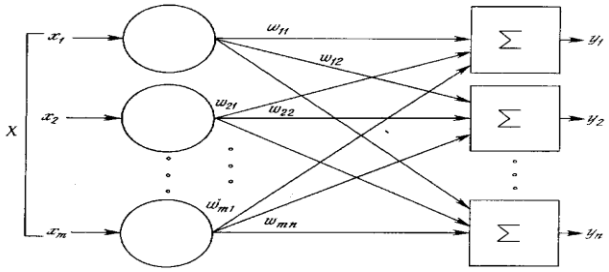

Figure 1. Single-layer neural network

The essence of this principle is as follows: for a given input vector XXX, the neuron in the first layer that produces the maximum (or minimum) value is considered to have "captured" the processed object. All properties of this neuron are then associated with that object. For instance, if the neurons in the layer are interpreted as representatives of different classes, the neuron that captures the object determines the class to which the object belongs. Consequently, a new (unknown) object is assigned to the same class as the neuron with the highest activation. A single-layer artificial neural network operating based on the maximum principle is illustrated in Figure 1.

Large-scale and complex neural networks typically possess correspondingly high computational capabilities. Although many neural network architectures have been developed, multilayer neural networks are often considered as simplified representations of certain layered structures of the human brain. Compared to single-layer neural networks, multilayer networks have significantly greater learning capacity. Currently, various algorithms have been developed for training multilayer neural networks, enabling them to solve more complex problems effectively.

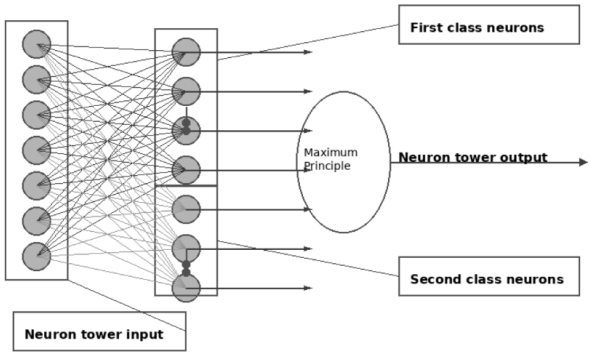

Figure 2. Single-layer artificial neural network operating based on the maximum principle.

Multilayer neural networks can be formed as a cascade of layers, where the output of one layer serves as the input to the next layer. Such a neural network structure is illustrated in Figure 2.

It should also be noted that, according to modern research, the equivalence between single-layer and multilayer neural networks has been mathematically proven by scholars in the field.

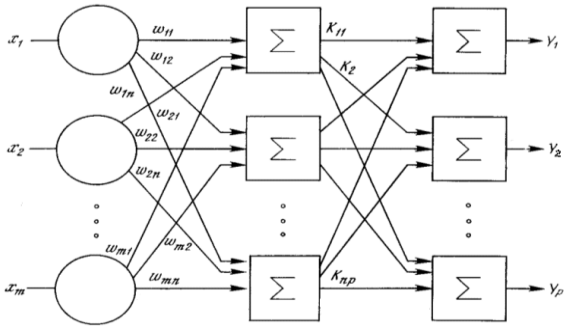

Figure 3. Two-layer neural network

Recurrent Neural Networks

The networks discussed above do not include feedback connections; that is, there are no links from the output of a given layer back to the input of the same or preceding layers. Such networks belong to the class of feedforward neural networks, which have attracted significant interest and are widely used in practice.

Networks in which outputs are connected back to inputs are referred to as recurrent neural networks. In networks without feedback connections, there is no memory, and their output depends solely on the current inputs and weights.

In certain types of recurrent neural networks, output values are fed back into the input, resulting in outputs that depend not only on current inputs but also on previous outputs. For this reason, recurrent neural networks exhibit properties similar to the short-term memory of the human brain. Thus, the output of such networks is partially dependent on previous inputs.

Presentation and Analysis of Simulation Results

The presentation and analysis of modeling results is one of the key aspects of simulation studies. To accomplish this, the modeling environment must support statistical data processing, structured or graphical data representation, and integration with external databases, among other functionalities. Specialized tools and software solutions can be employed to facilitate these processes.

Table 1. Stages of modeling

№	**Stage Name**	**Result**
1	System analysis	Understanding what occurs in the system, its structure, and the processes taking place within it
2	Formulation of modeling objectives	A list of tasks to be solved using the future model; a specification of input and output parameters of the model, a list of initial data, and the criteria for evaluating the performance of the tasks to be carried out.
3	Development of conceptual model structures	The structure of the model, the composition of key processes represented in the model, predefined levels for each subsystem (list of assumptions), and the description of control logic for the subsystems.
4	Modeling in the simulation environment	The implemented subsystems, their parameters and variables, their behavior, as well as the logic and interconnections of the subsystems.
5	Application of animation models	Visual representation of the model and user interface
6	Verification of the correctness of model implementation	This validated model is consistent with and accurately reflects the processes of the real system under analysis.
7	Model calibration	Distributions that reflect scenarios used for analyzing parameter values, equation coefficients, and models based on random variables.
8	Planning and conducting computer experiments	Simulation results presented in the form of graphs, tables, and other outputs.

In many cases, a simulation model is used as a module within a broader decision-making system that provides real-time monitoring of the controlled system, evaluates the consequences of the current situation, and suggests optimal (or reasonably efficient) control actions. To achieve this, it is typically necessary to integrate the model with other information systems and to develop a dedicated user interface.

A multilayer perceptron trained using the Hamming model algorithm can, in principle, be trained more efficiently if an appropriate activation function is applied. In most cases, this is implemented using strict threshold activation functions. However, the Hamming model also allows the use of other types of activation functions. In this study, the system is developed based on the classical Hamming model (see Figure 4).

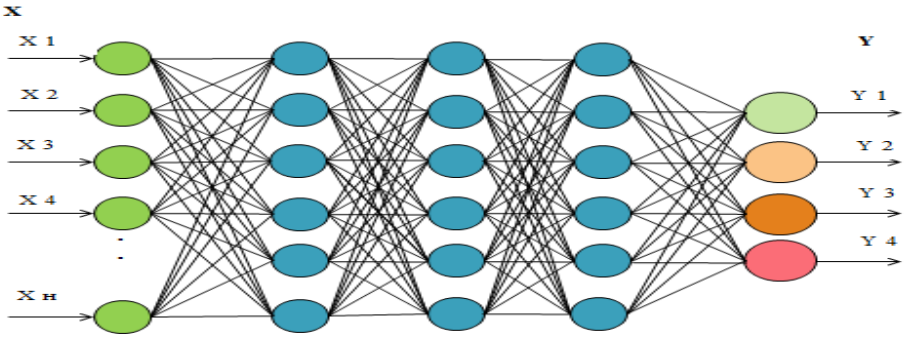

Figure 1. Structure of the classical Hamming model

Based on the selected model, DPI systems are designed, and the packet analysis capability of a network operating on the Hamming neural network model is evaluated. More accurate results could be achieved if this project is implemented using the NS-3 (Network Simulator), which is widely used in developed countries and supports real-world network configurations.

In our project, the system analyzes each packet through five stages. At the first stage, the incoming packet is examined based on its source IP address. This verification is performed within the first layer of the neural network. The greater the number of neurons in each layer, the higher the processing efficiency. At the second stage, the packet that has passed the initial check is analyzed based on its destination IP address. If the packet successfully passes this stage, it is then examined to determine whether it belongs to the UDP or TCP transport protocol. If the packet meets this requirement, it undergoes further inspection based on application-level signatures, which are defined separately for each service.

Project in the NS-3 simulator

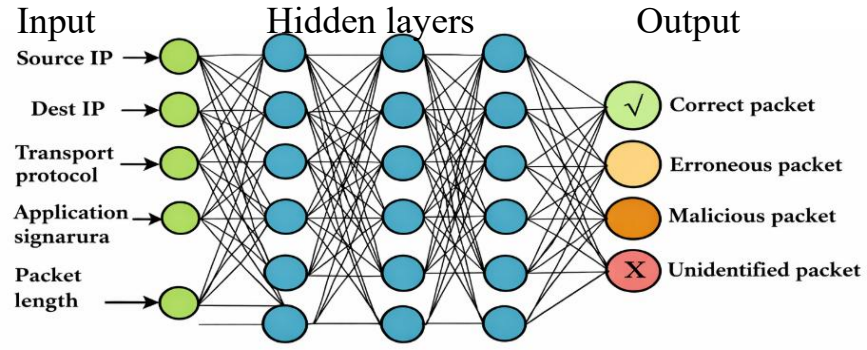

Figure 5. DPI system based on the Hamming model.

After determining which service the packet belongs to, it is necessary to measure its length. This is because each service has specific requirements regarding packet size. If a packet entering the network does not meet any of these requirements, it is immediately classified as malicious, infected, or unidentified and removed from the network accordingly.

To implement this project, four types of packets are generated using the C programming language. The first type fully complies with all specified requirements. The second type violates one of the requirements, specifically the standard packet length. The third type is defined as a malicious (virus-infected) packet. The fourth type represents an encrypted or unidentifiable packet.

These four types of packets are then introduced into the network as illustrated in Table 2.

Table 2. Number of packets entering the NS-3 system

Paket turi	Kirishdagi paketlar
To‘g‘ri	90000
Xato	5000
Xavfli	3000
Aniqlanmagan	2000
Jami	100000

Each neural network contains hidden layers, and each of these layers performs specific tasks (see Figure 6).

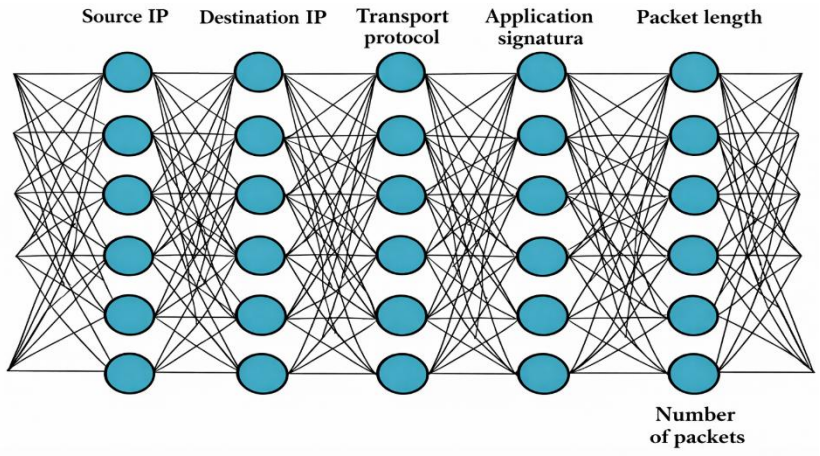

Figure 6. Hidden layers of the project

A total of 100,000 packets introduced into the network pass through five stages of verification and are stored in the cells of the sixth stage. At this stage, information such as the number of packets passing through each layer, the number and types of unidentified packets, and similar data are recorded.

These results allow us to observe how each packet enters the network, undergoes individual verification processes, and produces corresponding outcomes. The number of processed packets is then summarized and presented in the form of Table 1.

Table 1. Project results

Packet Type	Input Packets	Output Packets	Accuracy
Correct	90000	89931	99.92 %
Erroneous	5000	4992	99.84 %
Malicious	3000	2994	99.8 %
Unidentified	2000	2083	
Total	100000	100000	

The obtained results demonstrate that when DPI systems are designed based on the Hamming neural network model, out of 90,000 packets that meet the specified requirements, 89,931 packets were correctly identified. Among 5,000 erroneous packets, 4,992 were accurately classified as incorrect. Out of 3,000 malicious packets, 2,994 were successfully detected, while 83 packets that did not meet the requirements were classified as encrypted packets. The overall packet inspection accuracy was approximately 99%, which confirms the correctness and effectiveness of the proposed model and system design.

Conclusion

This study examined the characteristics of neural networks, including their structure, elements, and architectures, thereby highlighting their capabilities and advantages. Since neural networks operate on data, they are capable of continuous data analysis. Compared to conventional networks, neural networks offer higher speed, adaptability, and, most importantly, the ability to analyze data and predict workloads in advance.

The characteristics of neural network algorithms and models were analyzed, and the processing capabilities of Hamming and Hopfield models under network load conditions were evaluated. As a result, the Hamming model was selected as the most efficient approach, and DPI systems were designed based on this model. The results of the developed system demonstrate that the selected model performs effectively, achieving network analysis accuracy of up to 99%. These findings indicate that integrating neural networks into DPI systems significantly improves network performance and enhances overall system reliability.

Simulation results confirm that the proposed models function correctly and effectively filter network traffic. In the first model, the number of packets generated by the source did not match the number of packets received by the sink element. This discrepancy is due to the presence of packets from the fifth source that share similar port numbers and payload characteristics with packets from other sources, making them difficult to detect during the initial filtering stages. In the second model, signature-based filtering was incorporated, which resolved these issues. This improvement is clearly reflected in the simulation results of the second model.

Both simulation models demonstrate that network traffic filtering is essential, firstly, for resource optimization, secondly, for protecting service systems against threats and attacks, and thirdly, for reducing request processing time. As a result, both the speed and quality of service are significantly improved.

References

1. Kiminori Matsuzaki, Hideya Iwasaki, Kento Emoto, and Zhenjiang Hu. A library of constructive skeletons for sequential style of parallel programming. In Proceedings of the 1st international conference on Scalable information systems. ACM, 2006.
2. Johan Enmyren and Christoph W. Kessler. SkePU: a multi-backend skeleton programming library for multi-GPU systems. In Proceedings of the fourth international workshop on High-level parallel programming and applications. ACM, 2010.
3. А.О. Ключев, Д.Р. Ковязина, П.В. Кустарев, А.Е. Платунов: Аппаратные и программные средства встраиваемых систем. Санкт-Петербург. 2010
4. Worrall, A.; Carter, B.Widley, G.: Network monitor and method. 2008 [cit. 2014-03-06], uS Patent 7,411,946.
5. Matityahu, E.; Shaw, R.; Carpio, D.; aj.: Gigabits zero-delay tap and methods thereof. 2011 [cit. 2014-03-06], uS Patent App. 13/034,730.
6. Djuraev R. X., Baltaev J. B., Badalov J. I. Study of the method of compact testing of technical means of data transmission networks Toshkent //ICISCT2020.
7. Djurayev R. X., Baltayev J. B., Xasanov O. A. Increasing the efficiency of diagnosing microprocessor devices based on multichannel signal analysis means //2020 International Conference on Information Science and Communications Technologies (ICISCT). – IEEE, 2020. – С. 1-4.
8. Djuraev R. X., Djabbarov S. Y., Baltaev J. B. Sistemi texnicheskogo obslujivaniya i ekspluatatsii setey telekommunikatsii //Uchebnik.-T.:“Aloqachi. – 2019. – T. 234.
9. Djuraev R. H., Djabbarov S. Y., Baltayev J. B. Raqamli tizimlarning texnik diagnostikasi //Darslik).-T.:«Aloqachi. – 2020. – T. 232.