

МЕТОДИКА НЕЙРО-НЕЧЕТКОГО ПРОГНОЗИРОВАНИЯ ПОТЕРЬ ПАКЕТОВ ПРИ ПЕРЕГРУЗКЕ КОМПЬЮТЕРНОЙ СЕТИ**Польщикова К. А., Здоренко Ю. Н., Сокол Г. В.**

Рассмотрена научная задача, состоящая в разработке методики прогнозирования потерь пакетов при перегрузке компьютерной сети. Методика основана на построении и применении нечеткой нейронной сети для экстраполяции количества пакетов, которые будут отброшены маршрутизатором в течение ближайшего интервала времени. Обоснованы параметры функций принадлежности для входных величин, выбор алгоритма нечеткого вывода, алгоритма обучения и количества циклов обучения нейро-нечеткой системы. Показан пример создания и настройки нечеткой нейронной сети. Реализация предлагаемой в статье методики позволяет обеспечить достаточную точность прогноза потерь пакетов при перегрузке компьютерной сети.

Розглянута наукова задача, що полягає в розробці методики прогнозування втрат пакетів при перевантаженні комп'ютерної мережі. Методика ґрунтується на побудові й застосуванні нечіткої нейронної мережі для екстраполяції кількості пакетів, які будуть відкинута маршрутизатором протягом найближчого інтервалу часу. Обґрунтовані параметри функцій приналежності для входних величин, вибір алгоритму нечіткого виводу, алгоритму навчання і кількості циклів навчання нейро-нечеткої системи. Показано приклад створення і налаштування нечіткої нейронної мережі. Реалізація запропонованої в статті методики дозволяє забезпечити достатню точність прогнозування втрат пакетів при перевантаженні комп'ютерної мережі.

The article the decision of scientific task, consisting of development prognostication method of packet losses at the computer network overload is devoted. A method is based on a construction and application of unclear neuron network for extrapolation of packets amount which will be throw-away a router during the nearest time domain. The parameters of belonging functions are grounded for entrance sizes, choice of unclear conclusion algorithm, algorithm of teaching and amount of unclear neuron teaching cycles. The example of creation and tuning of unclear neuron network is offered. Realization of the method offered in the article allows providing sufficient exactness of packet losses prognostication at the computer network overload.

Польщикова К. А.

канд. техн. наук, доц. кафедры КИТ ДГМА
konspol@rambler.ru

Здоренко Ю. Н.

пом. нач. уч. отдела ВИТИ НТУУ «КПИ»
zdor@front.ru

Сокол Г. В.

канд. техн. наук, ст. преп. ПолтНТУ
Galochkapoltava@rambler.ru

ДГМА – Донбасская государственная машиностроительная академия, г. Краматорск

ВИТИ НТУУ «КПИ» – Военный институт телекоммуникаций и информатизации Национального технического университета Украины «Киевский политехнический институт», г. Полтава

ПолтНТУ – Полтавский национальный технический университет им. Юрия Кондратюка, г. Полтава

УДК 621.396

Польщиков К. А., Здоренко Ю. Н., Сокол Г. В.

МЕТОДИКА НЕЙРО-НЕЧЕТКОГО ПРОГНОЗИРОВАНИЯ ПОТЕРЬ ПАКЕТОВ ПРИ ПЕРЕГРУЗКЕ КОМПЬЮТЕРНОЙ СЕТИ

Перегрузки в компьютерной сети возникают в результате повышения интенсивности трафика в условиях ограниченной пропускной способности сетевых элементов. Вследствие перегрузок часть передаваемых по сети пакетов теряется. Прогнозирование величин, характеризующих потери пакетов в компьютерной сети, является важной задачей. Во-первых, зная уровень пакетных потерь, можно оценить масштаб возможных сетевых перегрузок. Во-вторых, данные о количестве потерянных пакетов на определенном временном интервале можно использовать для борьбы с перегрузками. И, в-третьих, величина утраченных пакетов характеризует качество доставки данных пользователям, т.к. анализируя ее значение, можно оценить задержки при передаче и долю потерянной при этом информации. Поэтому, исследования, связанные с прогнозированием потерь пакетов при перегрузке компьютерной сети, являются актуальными.

Эффективным средством, позволяющим успешно осуществлять экстраполяцию различных случайных функций, является применение гибридных интеллектуальных систем [1–3]. Одним из примеров таких систем является нечеткая нейронная сеть, сочетающая в себе преимущества нечеткой логики и искусственной нейронной сети.

Статья посвящена решению научной задачи, состоящей в разработке методики нейро-нечеткого прогнозирования потерь пакетов при перегрузке компьютерной сети.

Целью статьи является обеспечение точности предсказания количества пакетов, которые в течение ближайшего интервала времени будут отброшены маршрутизатором компьютерной сети.

Предположим интервал времени, в течение которого требуется осуществлять прогнозирование, условно разбит на участки одинаковой длительности. Известны значения количества пакетов, поступивших в выходной порт маршрутизатора, на любом из участков, находящихся до момента времени, соответствующего началу текущего участка.

Требуется предсказать, какое количество пакетов будет отброшено маршрутизатором на участке времени, который является последующим по отношению к текущему. Результат прогноза необходимо определять в начале каждого текущего участка.

Предлагаемая в данной статье методика включает процедуры *построения (синтеза) и функционирования* нечеткой нейронной сети для прогнозирования потерь пакетов при перегрузке компьютерной сети.

Синтез нечеткой нейронной сети предполагает последовательное выполнение указанных ниже этапов [4].

1. Определение входных и выходных величин нечеткой нейронной сети.
2. Выбор алгоритма нечеткого вывода.
3. Определение функций принадлежности для входных величин:
 - выбор количества функций принадлежности;
 - выбор формы функций принадлежности.
4. Выбор алгоритма обучения нечеткой нейронной сети.
5. Выбор количества циклов обучения нечеткой нейронной сети.
6. Создание обучающей выборки для настройки параметров нечеткой нейронной сети.
7. Настройка параметров нечеткой нейронной сети путем ее обучения.

Рассмотрим выполнение указанных этапов при построении нечеткой нейронной сети, которую предлагается использовать для прогнозирования количества пакетов, потерянных в результате перегрузки маршрутизатора.

Определение входных и выходных величин синтезируемой нечеткой нейронной сети.

Предположим, что текущим является участок e (см. рис. 1), тогда по отношению к нему предыдущими являются участки a , b , c и d , а последующим – участок f .

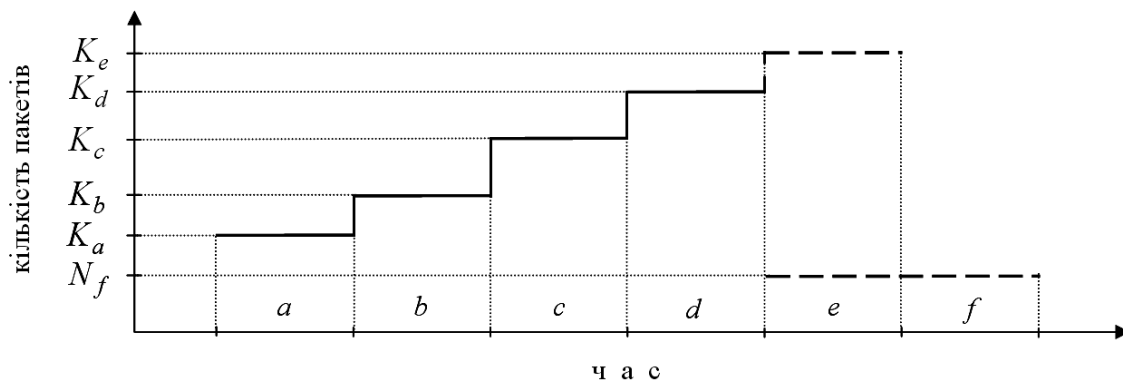


Рис. 1. Фрагмент разбиения интервала прогнозирования на участки времени

В соответствии с предлагаемой методикой в качестве входных величин нечеткой нейронной сети используются переменные K_a , K_b , K_c и K_d , указывающие, какое количество пакетов поступило в выходной порт маршрутизатора в течение каждого из четырех предыдущих участков времени. Выходной величиной синтезируемой нейро-нечеткой системы является переменная N_f – прогнозируемое значение отброшенных маршрутизатором пакетов на последующем участке времени.

Выбор алгоритма нечеткого вывода.

Для нечеткого вывода рекомендуется использовать алгоритм Сугено 0-го порядка [5]. Данный алгоритм является достаточно простым, и, вместе с тем, позволяет получать приемлемые результаты при прогнозировании.

Определение функций принадлежности для входных величин.

Из соображений максимального упрощения синтезируемой системы для каждого входного параметра рекомендуется использовать по две треугольные функции принадлежности. Благодаря этому структура нечеткой нейронной сети будет наименее сложной, а также снизится вычислительная сложность ее программной реализации.

Выбор алгоритма обучения нечеткой нейронной сети.

Для обучения нечеткой нейронной сети предлагается использовать алгоритм обратного распространения ошибки [6], являющийся одним из наиболее простых и распространенных.

Выбор количества циклов обучения нечеткой нейронной сети.

Количество циклов обучения не должно быть слишком малым, т. к. в этом случае отклонение реальных результатов прогнозирования от эталонных может быть недопустимо высоким. Выбор чрезмерно большого числа циклов обучения приводит к бесполезному выполнению многих итераций и замедляет процесс настройки параметров нечеткой нейронной сети. Исследования показали, что для успешной настройки синтезируемой системы достаточно выполнить 20 циклов обучения.

Создание обучающей выборки для настройки параметров нечеткой нейронной сети.

При формировании эталонных обучающих данных были использованы результаты измерения величины K_i – количества пакетов, поступивших в выходной порт маршрутизатора реальной компьютерной сети на наблюдаемых i -х участках времени ($0 \leq i \leq 100$), а также величины N_i – количества пакетов, отброшенных маршрутизатором на соответствующих участках времени (см. рис. 2 и рис. 3).

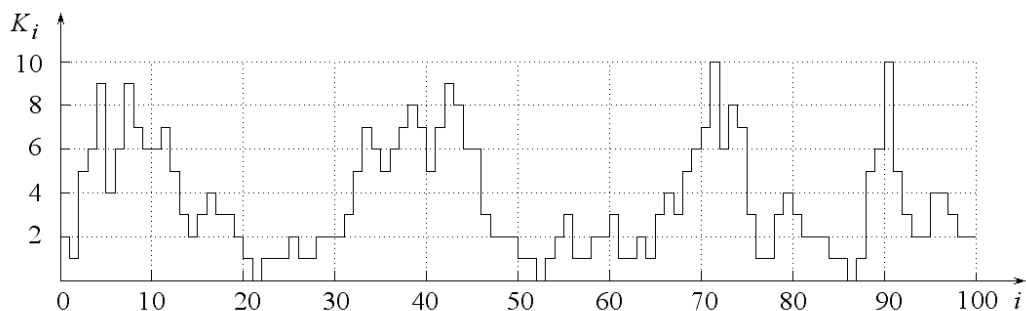


Рис. 2. Значения величины K_i

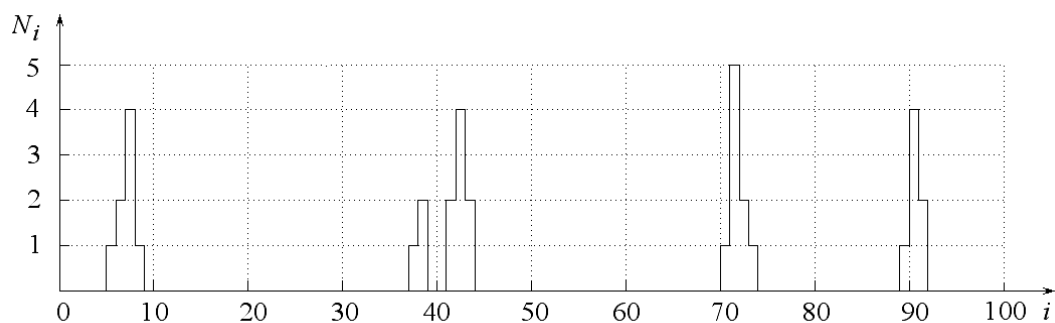


Рис. 3. Значения величины N_i

Фрагмент полученной матрицы данных для обучения нечеткой нейронной сети представлен в виде табл. 1.

Таблица 1

Фрагмент матрицы данных для обучения нечеткой нейронной сети

№ строки	Значения элементов матрицы				
	K_a	K_b	K_c	K_d	N_f
1.	2	1	5	6	1
2.	1	5	6	9	2
3.	5	6	9	4	4
...
37.	7	8	7	5	4
38.	8	7	5	7	2
39.	7	5	7	9	0
...
93.	2	2	4	4	0
94.	2	4	4	3	0
95.	4	4	3	2	0

Настройка параметров нечеткой нейронной сети путем ее обучения.

Синтез нечеткой нейронной сети осуществлялся в среде Matlab 6.5 [7]. Настройка параметров создаваемой системы производилась с использованием данных, содержащиеся в табл. 1, и параметров нечеткой нейронной сети, значения которых обоснованы выше при описании выполнения 2–5 этапов ее построения. В ходе настройки параметров данной системы получены выражения для функций принадлежности входных переменных:

$$\mu_1(K_a) = \begin{cases} 1, & \text{если } K_a < 0,008; \\ \frac{9,966 - K_a}{9,958}, & \text{если } 0,008 \leq K_a \leq 9,966; \\ 0, & \text{если } K_a > 9,966; \end{cases} \quad (1)$$

$$\mu_2(K_a) = \begin{cases} 0, & \text{если } K_a < 0,001; \\ \frac{K_a - 0,001}{9,991}, & \text{если } 0,001 \leq K_a \leq 9,992; \\ 1, & \text{если } K_a > 9,992; \end{cases} \quad (2)$$

$$\mu_1(K_b) = \begin{cases} 1, & \text{если } K_b < 2,045; \\ \frac{9,867 - K_b}{7,822}, & \text{если } 2,045 \leq K_b \leq 9,867; \\ 0, & \text{если } K_b > 9,867; \end{cases} \quad (3)$$

$$\mu_2(K_b) = \begin{cases} 0, & \text{если } K_b < 2,131; \\ \frac{K_b - 2,131}{7,909}, & \text{если } 2,131 \leq K_b \leq 10,04; \\ 1, & \text{если } K_b > 10,04; \end{cases} \quad (4)$$

$$\mu_1(K_c) = \begin{cases} 1, & \text{если } K_c < 2,013; \\ \frac{9,948 - K_c}{7,935}, & \text{если } 2,013 \leq K_c \leq 9,948; \\ 0, & \text{если } K_c > 9,948; \end{cases} \quad (5)$$

$$\mu_2(K_c) = \begin{cases} 0, & \text{если } K_c < 2,055; \\ \frac{K_c - 2,055}{7,955}, & \text{если } 2,055 \leq K_c \leq 10,01; \\ 1, & \text{если } K_c > 10,01; \end{cases} \quad (6)$$

$$\mu_1(K_d) = \begin{cases} 1, & \text{если } K_d < 1,97; \\ \frac{9,676 - K_d}{7,706}, & \text{если } 1,97 \leq K_d \leq 9,676; \\ 0, & \text{если } K_d > 9,676; \end{cases} \quad (7)$$

$$\mu_2(K_d) = \begin{cases} 0, & \text{если } K_d < 2,081; \\ \frac{K_c - 2,081}{7,89}, & \text{если } 2,081 \leq K_d \leq 9,971; \\ 1, & \text{если } K_d > 9,971; \end{cases} \quad (8)$$

а также значения индивидуальных выводов нечетких правил: $Y_1 = 0,999$, $Y_2 = -0,776$, $Y_3 = -1,714$, $Y_4 = 2,897$, $Y_5 = 3,85$, $Y_6 = 0,39$, $Y_7 = -0,669$, $Y_8 = -4,027$, $Y_9 = -2,912$, $Y_{10} = 5,769$, $Y_{11} = -1,233$, $Y_{12} = 8,498$, $Y_{13} = -0,178$, $Y_{14} = 5,771$, $Y_{15} = -1,132$, $Y_{16} = 0,321$.

В результате получена нейро-нечеткая система, структура которой приведена на рис. 4.

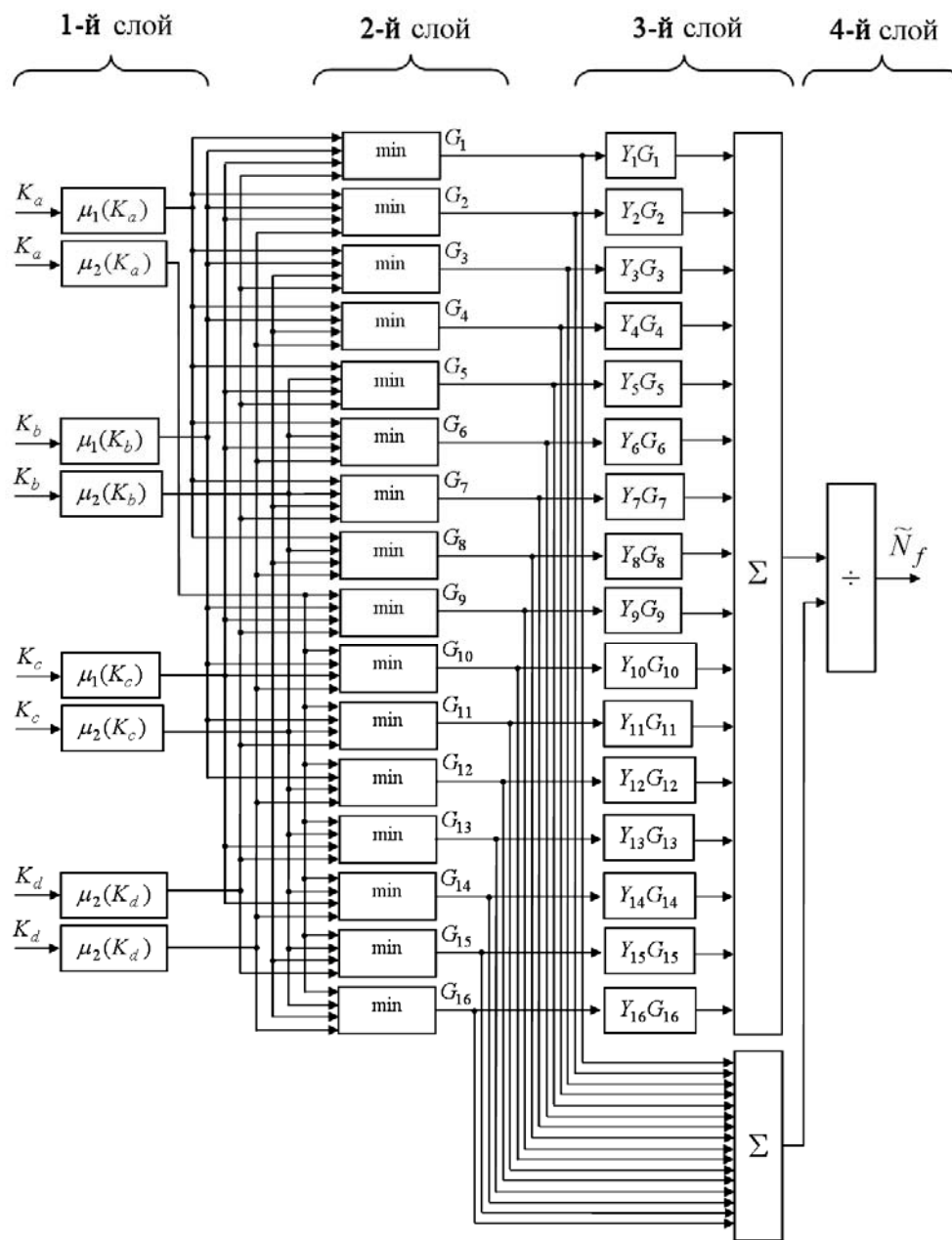


Рис. 4. Структура синтезированной нечеткой нейронной сети

Функционирование нечеткой нейронной сети основано на применении базы нечетких правил:

$$\left\{ \text{Если } (K_a = \alpha_1^1) \text{ и } (K_b = \alpha_2^1) \text{ и } (K_c = \alpha_3^1) \text{ и } (K_d = \alpha_4^1), \text{ то } (\tilde{N}_f = Y_1) \right\}, \quad (9)$$

$$\left\{ \text{Если } (K_a = \alpha_1^1) \text{ и } (K_b = \alpha_2^1) \text{ и } (K_c = \alpha_3^1) \text{ и } (K_d = \alpha_4^2), \text{ то } (\tilde{N}_f = Y_2) \right\}, \quad (10)$$

$$\left\{ \text{Если } (K_a = \alpha_1^1) \text{ и } (K_b = \alpha_2^1) \text{ и } (K_c = \alpha_3^2) \text{ и } (K_d = \alpha_4^1), \text{ то } (\tilde{N}_f = Y_3) \right\}, \quad (11)$$

$$\left\{ \text{Если } (K_a = \alpha_1^1) \text{ и } (K_b = \alpha_2^1) \text{ и } (K_c = \alpha_3^2) \text{ и } (K_d = \alpha_4^2), \text{ то } (\tilde{N}_f = Y_4) \right\}, \quad (12)$$

$$\left\{ \text{Если } (K_a = \alpha_1^1) \text{ и } (K_b = \alpha_2^2) \text{ и } (K_c = \alpha_3^1) \text{ и } (K_d = \alpha_4^1), \text{ то } (\tilde{N}_f = Y_5) \right\}, \quad (13)$$

$$\left\{ \text{Если } (K_a = \alpha_1^1) \text{ и } (K_b = \alpha_2^2) \text{ и } (K_c = \alpha_3^1) \text{ и } (K_d = \alpha_4^2), \text{ то } (\tilde{N}_f = Y_6) \right\}, \quad (14)$$

$$\left\{ \text{Если } (K_a = \alpha_1^1) \text{ и } (K_b = \alpha_2^2) \text{ и } (K_c = \alpha_3^2) \text{ и } (K_d = \alpha_4^1), \text{ то } (\tilde{N}_f = Y_7) \right\}, \quad (15)$$

$$\left\{ \text{Если } (K_a = \alpha_1^1) \text{ и } (K_b = \alpha_2^2) \text{ и } (K_c = \alpha_3^2) \text{ и } (K_d = \alpha_4^2), \text{ то } (\tilde{N}_f = Y_8) \right\}, \quad (16)$$

$$\left\{ \text{Если } (K_a = \alpha_1^2) \text{ и } (K_b = \alpha_2^1) \text{ и } (K_c = \alpha_3^1) \text{ и } (K_d = \alpha_4^1), \text{ то } (\tilde{N}_f = Y_9) \right\}, \quad (17)$$

$$\left\{ \text{Если } (K_a = \alpha_1^2) \text{ и } (K_b = \alpha_2^1) \text{ и } (K_c = \alpha_3^1) \text{ и } (K_d = \alpha_4^2), \text{ то } (\tilde{N}_f = Y_{10}) \right\}, \quad (18)$$

$$\left\{ \text{Если } (K_a = \alpha_1^2) \text{ и } (K_b = \alpha_2^1) \text{ и } (K_c = \alpha_3^2) \text{ и } (K_d = \alpha_4^1), \text{ то } (\tilde{N}_f = Y_{11}) \right\}, \quad (19)$$

$$\left\{ \text{Если } (K_a = \alpha_1^2) \text{ и } (K_b = \alpha_2^1) \text{ и } (K_c = \alpha_3^2) \text{ и } (K_d = \alpha_4^2), \text{ то } (\tilde{N}_f = Y_{12}) \right\}, \quad (20)$$

$$\left\{ \text{Если } (K_a = \alpha_1^2) \text{ и } (K_b = \alpha_2^2) \text{ и } (K_c = \alpha_3^1) \text{ и } (K_d = \alpha_4^1), \text{ то } (\tilde{N}_f = Y_{13}) \right\}, \quad (21)$$

$$\left\{ \text{Если } (K_a = \alpha_1^2) \text{ и } (K_b = \alpha_2^2) \text{ и } (K_c = \alpha_3^1) \text{ и } (K_d = \alpha_4^2), \text{ то } (\tilde{N}_f = Y_{14}) \right\}, \quad (22)$$

$$\left\{ \text{Если } (K_a = \alpha_1^2) \text{ и } (K_b = \alpha_2^2) \text{ и } (K_c = \alpha_3^2) \text{ и } (K_d = \alpha_4^1), \text{ то } (\tilde{N}_f = Y_{15}) \right\}, \quad (23)$$

$$\left\{ \text{Если } (K_a = \alpha_1^2) \text{ и } (K_b = \alpha_2^2) \text{ и } (K_c = \alpha_3^2) \text{ и } (K_d = \alpha_4^2), \text{ то } (\tilde{N}_f = Y_{16}) \right\}, \quad (24)$$

где α_1^A – терм (нечеткое множество) номер A входной переменной K_a , $A \in \overline{[1, 2]}$; α_2^B – терм номер B входной переменной K_b , $B \in \overline{[1, 2]}$; α_3^C – терм номер C входной переменной K_c , $C \in \overline{[1, 2]}$; α_4^D – терм номер входной переменной K_d , $D \in \overline{[1, 2]}$; Y_r – индивидуальный вывод нечеткого правила номер $r \in \overline{[1, 16]}$.

Как видно из рис. 4, созданная нейро-нечеткая система структурно состоит из четырех слоев. На вход этой системы в начале каждого текущего участка времени подаются значения входных переменных.

Первый слой нейронов выполняет процедуру фаззификации, состоящую в том, что для конкретных значений входных переменных K_a^* , K_b^* , K_c^* и K_d^* вычисляются значения соответствующих функций принадлежности $\mu_1(K_a^*)$, $\mu_2(K_a^*)$, $\mu_1(K_b^*)$, $\mu_2(K_b^*)$, $\mu_1(K_c^*)$, $\mu_2(K_c^*)$, $\mu_1(K_d^*)$ и $\mu_2(K_d^*)$.

Вторым слоем нейронов осуществляется агрегирование, в процессе которого используются выражения:

$$\left\{ G_1 = \mu_1(K_a^*) \wedge \mu_1(K_b^*) \wedge \mu_1(K_c^*) \wedge \mu_1(K_d^*) \right\}, \quad (25)$$

$$\{G_2 = \mu_1(K_a^*) \wedge \mu_1(K_b^*) \wedge \mu_1(K_c^*) \wedge \mu_2(K_d^*)\}, \quad (26)$$

$$\{G_3 = \mu_1(K_a^*) \wedge \mu_1(K_b^*) \wedge \mu_2(K_c^*) \wedge \mu_1(K_d^*)\}, \quad (27)$$

$$\{G_4 = \mu_1(K_a^*) \wedge \mu_1(K_b^*) \wedge \mu_2(K_c^*) \wedge \mu_2(K_d^*)\}, \quad (28)$$

$$\{G_5 = \mu_1(K_a^*) \wedge \mu_2(K_b^*) \wedge \mu_1(K_c^*) \wedge \mu_1(K_d^*)\}, \quad (29)$$

$$\{G_6 = \mu_1(K_a^*) \wedge \mu_2(K_b^*) \wedge \mu_1(K_c^*) \wedge \mu_2(K_d^*)\}, \quad (30)$$

$$\{G_7 = \mu_1(K_a^*) \wedge \mu_2(K_b^*) \wedge \mu_2(K_c^*) \wedge \mu_1(K_d^*)\}, \quad (31)$$

$$\{G_8 = \mu_1(K_a^*) \wedge \mu_2(K_b^*) \wedge \mu_2(K_c^*) \wedge \mu_2(K_d^*)\}, \quad (32)$$

$$\{G_9 = \mu_2(K_a^*) \wedge \mu_1(K_b^*) \wedge \mu_1(K_c^*) \wedge \mu_1(K_d^*)\}, \quad (33)$$

$$\{G_{10} = \mu_2(K_a^*) \wedge \mu_1(K_b^*) \wedge \mu_1(K_c^*) \wedge \mu_2(K_d^*)\}, \quad (34)$$

$$\{G_{11} = \mu_2(K_a^*) \wedge \mu_1(K_b^*) \wedge \mu_2(K_c^*) \wedge \mu_1(K_d^*)\}, \quad (35)$$

$$\{G_{12} = \mu_2(K_a^*) \wedge \mu_1(K_b^*) \wedge \mu_2(K_c^*) \wedge \mu_2(K_d^*)\}, \quad (36)$$

$$\{G_{13} = \mu_2(K_a^*) \wedge \mu_2(K_b^*) \wedge \mu_1(K_c^*) \wedge \mu_1(K_d^*)\}, \quad (37)$$

$$\{G_{14} = \mu_2(K_a^*) \wedge \mu_2(K_b^*) \wedge \mu_1(K_c^*) \wedge \mu_2(K_d^*)\}, \quad (38)$$

$$\{G_{15} = \mu_2(K_a^*) \wedge \mu_2(K_b^*) \wedge \mu_2(K_c^*) \wedge \mu_1(K_d^*)\}, \quad (39)$$

$$\{G_{16} = \mu_2(K_a^*) \wedge \mu_2(K_b^*) \wedge \mu_2(K_c^*) \wedge \mu_2(K_d^*)\}. \quad (40)$$

С помощью третьего слоя сети выполняется активизация, в процессе которой вычисляется сумма $\sum_{r=1}^{\rho} Y_r$ и взвешенная сумма $\sum_{r=1}^{\rho} Y_r G_r$ выходных сигналов второго слоя.

Четвертый слой нейронов осуществляет дефаззификацию, т. е. определение четкого значения выходной переменной по формуле:

$$\tilde{N}_f = \frac{\sum_{r=1}^{\rho} Y_r G_r}{\sum_{r=1}^{\rho} G_r}. \quad (41)$$

Рассмотрим результаты одного из имитационных экспериментов, проведенного с применением полученной нечеткой нейронной сети (см. рис. 5–10).

Рис. 5 иллюстрирует фрагмент процесса поступления пакетов в выходной порт маршрутизатора для дальнейшей отправки по компьютерной сети. Каждый поступающий пакет

изображен в виде короткого прямоугольного импульса, амплитуда которого k_n соответствует порядковому номеру этого пакета. Значения i соответствуют номерам эквидистантных участков времени, на которые разбивается анализируемый процесс. Текущие значения количества пакетов K_i , поступивших в выходной порт маршрутизатора, приводятся на рис. 6. Динамика изменения текущей длины пакетной очереди Q выходного порта маршрутизатора представлена на рис. 7. Максимальный размер указанной очереди ограничен битовой длиной шестнадцати пакетов. Поэтому потери пакетов наблюдаются на участках с номерами 6, 7 и 8 (см. рис. 8), на которых $Q=16$. Номера отброшенных пакетов соответствуют значениям k_o . Реальные значения количества отброшенных пакетов N_i , полученные на каждом текущем участке времени, можно наблюдать на рис. 9. Прогнозируемые потери пакетов \tilde{N}_i , получаемые с помощью созданной нечеткой нейронной сети в начале каждого текущего участка времени, представлены на рис. 10. Видно, что полученные результаты прогноза практически совпадают с реальными потерями пакетов на соответствующих участках времени, что свидетельствует об адекватности предсказания.

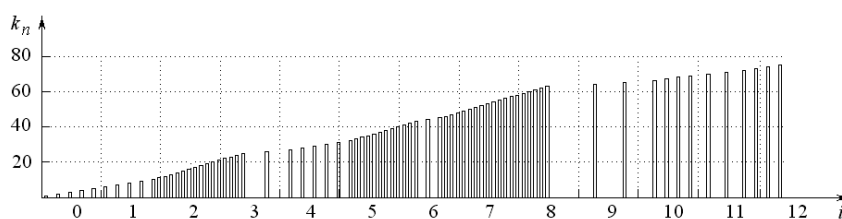


Рис. 5. Номера пакетов, поступивших в выходной порт маршрутизатора для дальнейшей отправки по компьютерной сети

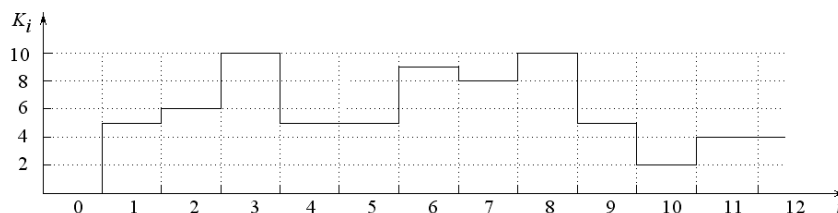


Рис. 6. Значения величины K_i

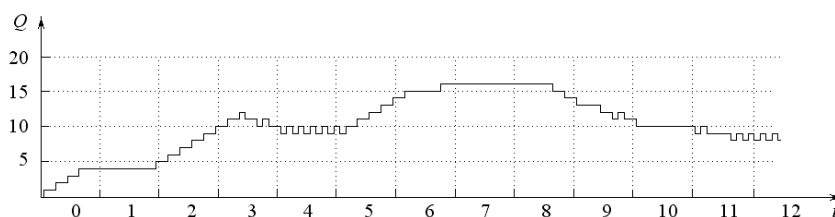


Рис. 7. Динамика изменения величины Q

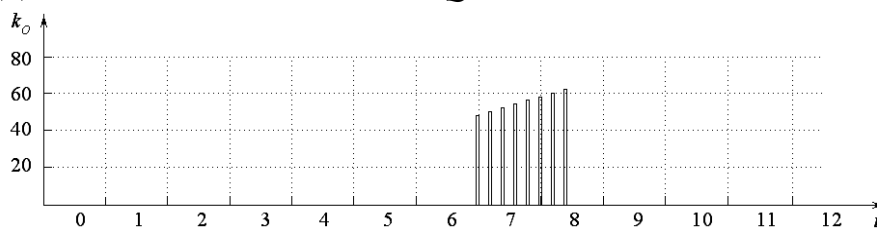


Рис. 8. Номера пакетов, отброшенных при перегрузке

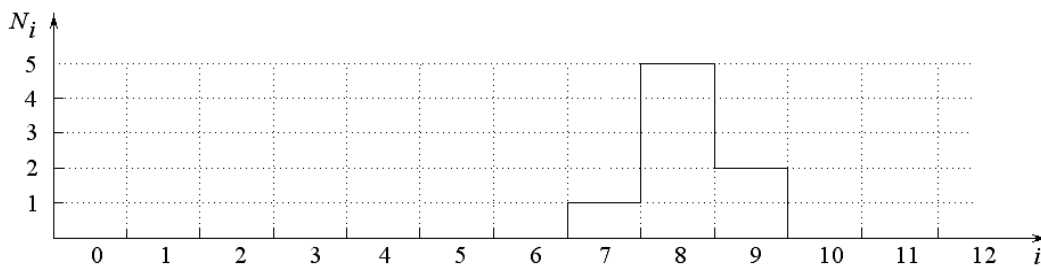


Рис. 9. Значения величины N_i

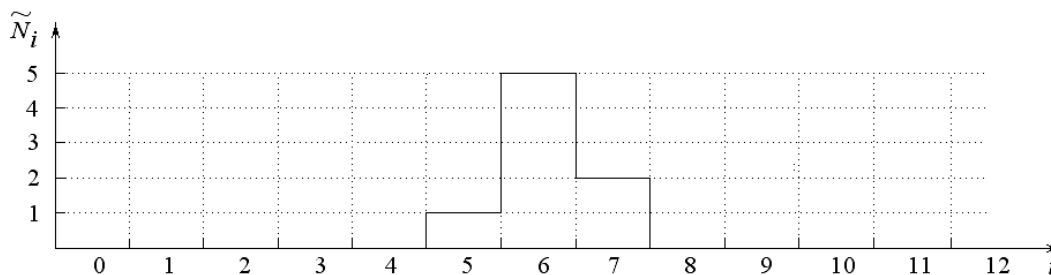


Рис. 10. Значения величины \tilde{N}_i

Для оценки точности предсказания количества пакетов, которые на протяжении ближайшего интервала времени будут отброшены маршрутизатором компьютерной сети, проведено более сотни экспериментов для различных реализаций случайной функции $k_n(t)$. В результате величина максимального отклонения реальных и прогнозируемых потерь пакетов не превышает $\pm 2\%$. Таким образом, применение предложенной в данной статье методики обеспечивает достаточную точность прогнозирования потерь пакетов при перегрузке компьютерной сети.

ВЫВОДЫ

Разработана методика прогнозирования потерь пакетов при перегрузке компьютерной сети, которая включает процедуры синтеза и функционирования соответствующей нейро-нечеткой системы. Величина максимального отклонения реальных и прогнозируемых потерь пакетов, получаемая при реализации предложенной методики, не превышает $\pm 2\%$.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Усков А. А. *Интеллектуальные технологии управления. Искусственные нейронные сети и нечеткая логика* / А. А. Усков, А. В. Кузьмин. – М. : Горячая линия – Телеком, 2004. – 143 с.
2. Гавилов А. В. *Гибридные интеллектуальные системы* / А. В. Гавилов. – Новосибирск : Изд-во НГТУ, 2002. – 142 с.
3. Польщиков К. О. *Метод управління тайм-аутом повторної передачі в телекомунікаційній мережі військового призначення на основі використання системи нейро-нечіткого виводу* / К. О. Польщиков // *Збірник наукових праць*. – К. : ВІПІ НТУУ «КПІ», 2008. – Вип. 3. – С. 98–108.
4. Леоненков А. В. *Нечеткое моделирование в среде MATLAB и fuzzyTECH* / А. В. Леоненков. – СПб. : БХВ-Петербург, 2003. – 703 с.
5. Takagi T. *Fuzzy identification of systems and its applications to modeling and control* / T. Takagi, M. Sugeno // *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 15, no. 1, 1985. – P. 116–132.
6. Rumelhart D. E. *Learning internal representations by error propagation, in parallel distributed procesing, t. 1* / D. E. Rumelhart, G. E. Hinton, R. J. Williams Eds. – Cambridge, MA, M.I.T. Press, 1986.
7. Дьяконов В. П. *MATLAB 6/6.1/6.5 + Simulink 4/5. Основы применения. Полное руководство пользователя* / В. П. Дьяконов. – М. : СОЛОН-Пресс, 2002. – 768 с.