

Ю.А. Леохин

ПРОГНОЗИРОВАНИЕ СОСТОЯНИЙ КОРПОРАТИВНОЙ СЕТИ ЦЕНТРА УПРАВЛЕНИЯ ПОЛЕТАМИ В НЕЙРО- СЕТЕВОМ ЛОГИЧЕСКОМ БАЗИСЕ

В статье обосновывается выбор типа нейронной сети, описывается структура и алгоритм подсистемы прогнозирования состояния корпоративной сети ЦУПа.

Ключевые слова: центр управления полетами, локальная вычислительная сеть, нейронная сеть, нейросетевой логический базис, прогнозирование, искусственный интеллект

Yu.L. Leokhin

PREDICTION OF STATE CORPORATE NETWORK MISSION CONTROL CENTER IN NEURAL NETWORK LOGICAL BASIS

The article proves the choice of type of neural network, describes the structure and algorithm of subsystem for forecasting state of the corporate network of MCC.

Keywords: mission control center, local area network, a neural network, neural network logical basis, forecasting, artificial intelligence

Введение

Российский Центр управления полетами (ЦУП) Федерального Космического Агентства (Роскосмос) – наиболее крупное научно-исследовательское подразделение ФГУП «Центральный научно-исследовательский институт машиностроения». Из ЦУПа осуществляется управление полетами космических аппаратов научного и народнохозяйственного назначения: Международной космической станцией, транспортными пилотируемыми кораблями «Союз» и грузовыми – «Прогресс», космическими аппаратами дистанционного зондирования Земли «Океан-О», «Метеор – 3М», «Фотон», «Компас», «Ресурс-ДК», разгонным блоком ДМ программы «Морской старт».

В 1993 году перед ЦУП встала задача коренной модернизации его информационно-вычислительной системы (ИВС), построенной на базе ВС Эльбрус, в связи увеличением объема работ в рамках международного сотрудничества, ростом сложности решаемых задач и моральным устареванием вычислительного оборудования [13].

Новая ИВС ЦУП объединила вычислительные средства, систему телекоммуникаций, локальные и глобальную вычислительные сети Центра.

В состав ИВС ЦУП входят Телеметрический, Баллистический и Командный информационно-вычислительные комплексы (ИВК). Основу парка вычислительных средств этих комплексов составили около 30 серверов и более 200 рабочих станций HP на основе Intel- и RISC-процессоров. Важнейшее оборудование работает в режиме горячего резервирования, обеспечивая высокую отказоустойчивость и надежность комплексов.

Здания ЦУП соединены высокоскоростными оптоволоконными каналами, внутри зданий используются структурированные кабельные системы. Основным каналобразующим и коммутационным оборудованием являются маршрутизаторы и коммутаторы компании Cisco Systems.

ЦУП осуществляет обмен информацией с американскими (Техас, Алабама, Вашингтон и др.) и европейскими (во Франции и Германии) космическими центрами, а также с отечественными центрами обработки научной информации и постановщиками космических экспериментов Российской академии наук. Этот обмен выполняется через комплекс внешних информационных обменов, основными элементами которого являются маршрутизаторы, сетевые серверы и персональные компьютеры HP и Cisco Systems.

Таким образом, корпоративная сеть ЦУПа представляет собой сложный информационно-технический объект, требующий прозрачного и эффективно управления. Важной функцией управления является прогнозирование состояний сети ЦУПа. В настоящее время для прогноза состояний сетей среды используются такие интегральные показатели качества функционирования телекоммуникационной сети, как пропускная способность, реакция на характеристики профиля трафика, количество потерянных пакетов, время доставки, неравномерность доставки пакетов.

Однако прогноз, составленный по приведенным выше интегральным показателям, будет недостаточно точным. Поэтому предлагается контролировать сеть ЦУП, т.е. выполнять мониторинг, анализ и прогнозирование, по значениям объектов, хранящихся в базах МІВ устройств [4]. В статье обосновывается

выбор типа нейронной сети, описывается структура и алгоритм подсистемы прогнозирования состояния корпоративной сети ЦУПа.

Подсистема прогнозирования состояния корпоративной сети NeuroNet

В настоящее время существует большое множество коммерческих пакетов, решающих задачи прогнозирования. Например, многофункциональный пакет Statistica (версия 6.0) [7]. В данном пакете представлены стандартные средства для решения задач прогнозирования, такие как моделирование нейронной сети, генерация различных графиков, оценка чувствительности параметров и т.д. Но поскольку средства являются стандартными, не всегда есть возможность сделать сеть достаточно гибкой для решения конкретной задачи. Кроме этого, интеграция коммерческого программного обеспечения в разработанную структуру системы управления корпоративной сетью является достаточно сложной задачей. Поэтому возникла необходимость в разработке подсистемы прогнозирования состояния корпоративной сети.

При выборе типа нейронной сети были рассмотрены следующие нейронные сети [1, 3, 5, 6, 8, 9, 10, 11, 12]: персептрон, многослойный персептрон, радиально-базисные сети, линейные сети, сети Ворда. Сравнительный анализ этих типов позволил обосновать выбор многослойного персептрона, который наиболее полно отвечает поставленной задаче прогноза состояния корпоративной сети. Выбор проводился по следующим критериям:

1. **Критерий существования.** Исходя из следствия теоремы Колмогорова [3], для решения поставленной задачи всегда существует многослойный персептрон, с помощью которого можно производить экстраполяцию выхода многослойного персептрона со сколь угодно заданной точностью.

2. **Критерий времени.** Производительность многослойного персептрона соизмерима с процедурой перемножения матриц (в зависимости от количества слоев). Исходя из производительности современных компьютеров, можно считать, что при прогнозировании реализуем режим «реального» времени.

3. **Критерий гибкости настроек и реализации.** Многослойный персептрон в зависимости от требований поставленной задачи позволяет менять свою конфигурацию и настроечные параметры.

Для того чтобы определить необходимое количество нейронов в слое и количество слоев в сети необходимо проводить множество экспериментов, анализируя показатели которых можно определить, насколько та или иная конфигурация отвечает поставленным требованиям. Аналитически опреде-

лить эти параметры невозможно, поэтому применяется экспериментальный подход. Были рассмотрены различные конфигурации, и в качестве оптимальной была принята (24, 15, 10, 7), то есть 24 входных параметра, 15 и 10 нейронов в скрытых слоях и 7 выходных нейронов. Таким образом, была выбрана трехслойная конфигурация персептрона.

Подсистема разработана на объектно-ориентированном языке программирования C++ и с использованием базы данных MySQL.

Структура и алгоритм подсистемы прогнозирования состояния корпоративной сети NeuroNet

Подсистема прогнозирования состояния корпоративной сети NeuroNet состоит из трех основных компонент: Графический интерфейс, Нейронная сеть и База данных (рис. 1).

Графический интерфейс позволяет отображать на экране всю необходимую информацию, а также задавать переменные, необходимые для построения нейронной сети. Он включает следующие модули:

- модуль инициализации переменных, позволяющий задавать и изменять различные параметры, необходимые для построения нейронной сети;
- модуль отображения полученных результатов, позволяющий наглядно отображать результаты работы программного комплекса.

База данных отвечает за работу с базой данных MySQL и содержит следующие модули:

- модуль подключения к базе данных, отвечающий за доступ к серверу, где хранится вся информация по базам данных МІВ;
- модуль хранения и изменения данных, отвечающий за возможность сохранять новые данные, а также изменять (удалять) уже имеющиеся;
- модуль выборки данных позволяет, в зависимости от выбранных параметров модуля инициализации переменных, отбирать только необходимые для построения нейронной сети данные.

Нейронная сеть позволяет создавать, тестировать и анализировать различные конфигурации нейронной сети, а также загружать и работать с уже имеющейся нейронной сетью. Включает следующие модули:

- модуль предварительной обработки данных, отвечающий за приведение входных данных в формат, необходимый для подачи на вход нейронной сети;
- модуль формирования параметров новой нейронной сети отвечает за инициализацию параметров, необходимых для создания новой нейронной сети, за вычисление локального поля нейрона, выходного сигнала нейрона и ошибки для нейро-

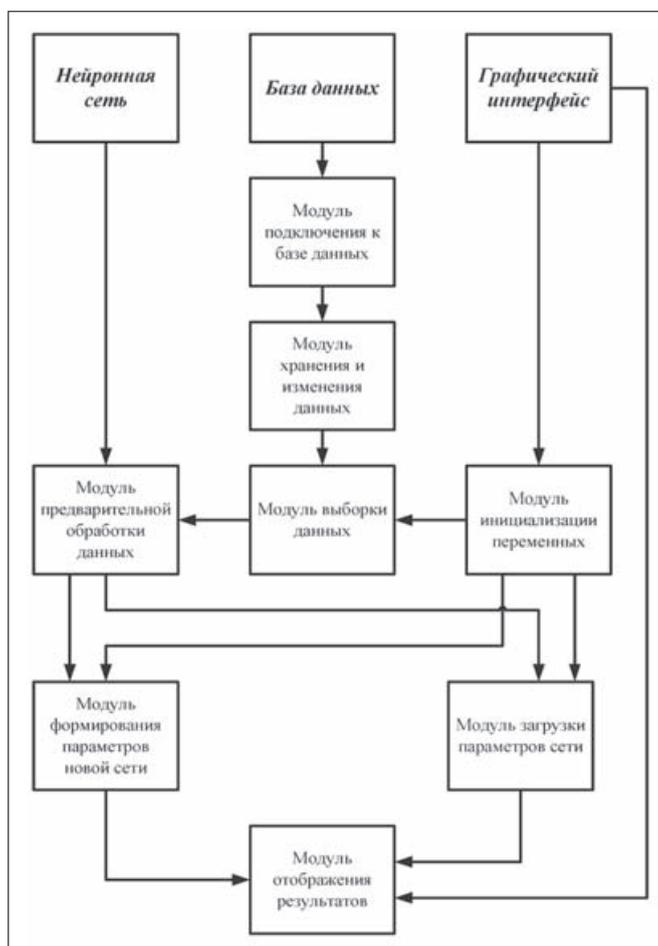


Рис. 1. Структура подсистемы прогнозирования состояния корпоративной сети

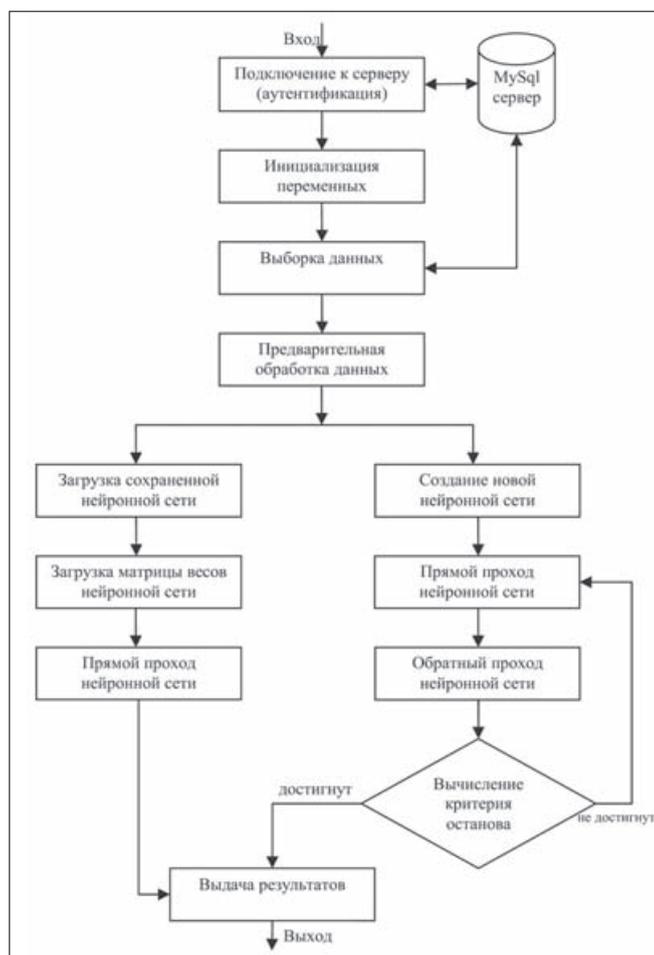


Рис. 2. Схема алгоритма функционирования подсистемы прогнозирования NeuroNet

нов выходного слоя, за вычисление локальных градиентов узлов и коррекцию синоптических весов нейронов;

- модуль загрузки параметров сети отвечает за реализацию возможности загрузить уже сохраненную конфигурацию сети для дальнейшей работы с ней и за вычисление выходного сигнала нейрона.

Процедуры прямого и обратного прохода осуществляются в цикле до достижения критерия останова. После этого вся информация подается в модуль «Отображение полученных результатов».

Алгоритм функционирования программного комплекса состоит из 8 этапов (рис. 2):

Подключение к серверу. На данном этапе, для того чтобы получить возможность работать с базой данных, получать и обрабатывать информацию, необходимо пройти аутентификацию на сервере. После чего вся информация, находящаяся на сервере, будет доступна для изменения.

Инициализация переменных. На данном этапе инициализируются все необходимые для работы

переменные, такие как количество эпох, время обработки, синаптические веса, пороговые значения и т.д. Таким образом, можно добиться того, что нейронная сеть будет полностью отвечать запросам пользователя. Возможность самостоятельно выбирать количество нейронов в слое позволяет реализовать гибкость системы в зависимости от потребностей.

Выборка данных. На данном этапе после определения ряда переменных количество необходимых наборов данных сокращается. Таким образом, на вход сети будет подаваться не хаотичный набор значений, а только строго отобранные переменные. Используя метод главных компонент, были отобраны только те данные, которые представляют интерес и несут в себе информацию по состоянию сети на данный момент времени. Увеличивая или уменьшая период выборки данных можно рассматривать состояние сети в различные моменты времени, соизмеримые с ним.

Предварительная обработка данных. На данном этапе выполняется обработка данных, полученных из

серверной части, в формат, требуемый для входа нейронной сети. Производится нормализация параметров. Предварительная обработка данных осуществляется с помощью вычисления среднего значения временного ряда и среднеквадратичного отклонения в соответствии с выражениями (1), (2) и (3) [1, 2, 3]:

$$1. \bar{x}_i = \frac{1}{P} \sum_{\alpha=1}^P x_i^\alpha, \quad (1)$$

где \bar{x} – средневыворочное временного ряда; P – количество членов временного ряда; $\alpha = \overline{1, P}$, x_i^α – значение i -го параметра МВ-устройства корпоративной сети в момент времени α .

$$\sigma_i^2 \equiv \frac{1}{P-1} \sum_{\alpha=1}^P (x_i^\alpha - \bar{x}_i)^2, \quad (2)$$

где σ_i – среднее квадратичное отклонение; x_i – i -ое нормированное значение временного ряда.

$$\tilde{x}_i = \left(\frac{x_i - \bar{x}_i}{\sigma_i} \right). \quad (3)$$

2. Создание нейронной сети. На данном этапе происходит инициализация параметров, необходимых для создания новой нейронной сети, и непосредственное создание сети заданной конфигурации. Также происходит отборка данных, необходимых для ее обучения, и собственно процесс обучения.

3. Загрузка нейронной сети. Этот этап отвечает за возможность загрузки в систему ранее сохраненной конфигурации сети. После загрузки конфигурации сети на основе выбранных данных можно проводить прогнозирование и предсказание.

4. Прямой проход нейронной сети. На данном этапе осуществляется расчет локального поля нейрона по следующему выражению (4) [1, 2, 3]:

$$v_j^{(l)}(n) = \sum_{i=0}^{m_0} w_{ji}^{(l)}(n) y_i^{(l-1)}(n) \quad (4)$$

Здесь: $v_j^{(l)}$ – локальное поле выхода нейрона j , находящегося в слое l в момент времени n ; $w_{ji}^{(l)}(n)$ – весовой коэффициент синаптической связи i -го нейрона слоя l с j -ым нейроном слоя $l-1$ в момент времени n ; $y_i^{(l-1)}(n)$ – выход i -го нейрона слоя $l-1$ в момент времени n .

Вычисление выходного сигнала нейрона осуществляется согласно выражению (5):

$$y_j^{(l)}(n) = \varphi_j \left(v_j^{(l)}(n) \right), \quad (5)$$

где $y_j^{(l)}$ – выход j -ого нейрона слоя l в момент времени n ; $\varphi_j^{(l)}(v_j^{(l)}(n))$ – функция активации j -ого нейрона слоя l .

Вычисление сигнала ошибки для нейронов выходного слоя производится по формуле (6):

$$e_j(n) = d_j(n) - y_j(n), \quad (6)$$

где $e_j(n)$ – ошибка для j -ого нейрона в момент времени n ; $d_j(n)$ – желаемый выход j -ого нейрона в момент времени n ; $y_j(n)$ – полученное значение выхода j -ого нейрона в момент времени n .

Обратный проход. На данном этапе осуществляется расчет локальных градиентов узлов для нейронов скрытых и выходных слоев [5, 6, 8, 9] по формуле (7):

для нейрона выходного слоя,

$$\delta_j^{(l)}(n) = \begin{cases} e_j^{(L)}(n) \varphi_j^{(L)} \left(v_k^{(L)}(n) \right) \\ \varphi_j^{(L)} \left(v_j^{(L)}(n) \right) \sum_k \delta_k^{(l+1)}(n) w_{kj}^{(l+1)}(n) \end{cases} \quad (7)$$

для нейрона скрытого слоя.

Здесь: $\delta_j^{(l)}(n)$ – локальный градиент для j -ого нейрона слоя l .

Также происходит корректировка синаптических весов по формуле (8):

$$\Delta w_{ji}^{(l)}(n) = \alpha \Delta w_{ji}^{(l)}(n-1) + \eta \delta_j^{(l)}(n) y_i^{(l-1)}(n), \quad (8)$$

где α – коэффициент поправочный, равный 0,65; $\Delta w_{ji}^{(l)}(n)$ – изменение весового коэффициента синаптической связи; η – коэффициент момента, равный 0,7.

Выбор системы информативных переменных

Большинство администраторов для прогнозирования используют небольшое количество интегральных параметров – объем трафика, задержки в узлах коммутации, количество потерянных пакетов. Анализ небольшого количества параметров не позволяет получить качественный прогноз и подходит для крупных магистралей сетей. Для повышения качества прогноза предлагается использовать значения параметров МВ устройств. В корпоративной сети

Таблица 1. Входные переменные

<code>interface.iftable.ifentry.IflnUcastPkts</code>	Число пакетов, доставленных на верхний системный уровень (unicast)
<code>interface.iftable.ifentry.IflnNucastPkts</code>	Число пакетов, доставленных на верхний системный уровень (nonunicast)
<code>interface.iftable.ifentry.IfOutUcastPkts</code>	Число unicast-пакетов, полученных с верхнего системного уровня
<code>interface.iftable.ifentry.IfOutNucastPkts</code>	Число мультикастинг- и широковещательных пакетов, полученных с верхнего системного уровня
<code>interface.iftable.ifentry.IflnDiscads</code>	Число полученных, но отвергнутых пакетов
<code>interface.iftable.ifentry.IflnUnknownProtos</code>	Число полученных пакетов с неизвестным кодом протокола
<code>interface.iftable.ifentry.IfOutDiscads</code>	Количество отвергнутых пакетов из числа отправленных
<code>ip.IplnUnknownProtos</code>	Число входных IP-дейтограмм, с кодами протоколов, которые не поддерживаются данной системой
<code>ip.IplnReceives</code>	Полное число IP-дейтограмм, включая полученные с ошибкой
<code>ip.IpReasmReqds</code>	Число полученных фрагментов, которые требуют сборки
<code>ip.IplnDelivers</code>	Число IP-дейтограмм, принятых без ошибок (включая ICMP)
<code>ip.ipReasmOKs</code>	Число полученных и успешно собранных IP-дейтограмм
<code>ip.IplnHdrErrors</code>	Число входных IP-дейтограмм с ошибками в заголовке пакета, включая ошибки контрольной суммы, TTL и т.д.
<code>icmp.IcmplnMsgs</code>	Число полученных ICMP-пакетов
<code>icmp.IcmpOutDestUnreachs</code>	Число ICMP-сообщений о недоступности адресата
<code>icmp.IcmpOutSrcQuench</code>	Число посланных ICMP-сообщений об уменьшении потока пакетов
<code>icmp.IcmpOutMsgs</code>	Число отправленных ICMP-сообщений
<code>udp.UdplnDatagrams</code>	Число принятых UDP-дейтограмм
<code>udp.UdpOutDatagrams</code>	Число отправленных UDP-дейтограмм
<code>udp.UdpNoPorts</code>	Полное число UDP-дейтограмм, где не существует приложения для указанного номера порта
<code>tcp.TcplnSegs</code>	Число принятых TCP-сегментов
<code>tcp.TcpOutSegs</code>	Число отправленных TCP-сегментов
<code>tcp.TcpRetransSegs</code>	Число tcp-сегментов с повторной пересылкой
<code>tcp.TcpOutRsts</code>	Число сегментов с флагом RST=1

Таблица 2. Выходные переменные

<code>interface.iftable.ifentry.IflnErrors</code>	Число ошибок при приеме пакетов
<code>interface.iftable.ifentry.IfOutErrors</code>	Число отправленных пакетов, содержащих ошибки
<code>ip.IplnAddrErrors</code>	Число полученных пакетов с ошибкой в адресе
<code>icmp.IcmplnErrors</code>	Число ICMP-сообщений, полученных с ошибками
<code>icmp.IcmpOutErrors</code>	Число не отправленных ICMP-сообщений из-за проблем в ICMP (напр. нехватка буферов)
<code>udp.UdplnErrors</code>	Число UDP-дейтограмм, которые не могут быть доставлены по причине отсутствия приложения по указанному порту
<code>tcp.TcplnErrs</code>	Число TCP-сегментов, полученных с ошибкой

это возможно, так как все ресурсы принадлежат предприятию, и доступ к устройствам и сетевым узлам у администраторов не ограничен.

Были отобраны 31 переменная: 24 входных и 7 выходных переменных. Они описаны в таблицах 1 и 2.

Выбор определялся тем, что именно эти переменные варьируются динамически в течение суток. При разработке подсистемы прогнозирования решалась задача исследования временных вариаций потоков в

заданном сегменте и выработки критериев для диагностики потенциально опасных ситуаций. Измерения проводились каждый час в течение нескольких месяцев.

Многие переменные базы MIB не менялись или менялись незначительно, но определяли режим работы и состояние сервера. Так, переменная *snmpinbadcommunityuses {snmp 5}* может сообщить о попытках несанкционированного доступа к базе MIB.

Переменные `snmpintoobigs {snmp 8}`, `snmpingenerrs {snmp 12}` или `ifadminstatus {ifentry 7}` и некоторые другие характеризуют текущее состояние системы, и длительное их отслеживание чаще всего не даст полезной информации. Другие переменные, например, `ipnettomedianetaddress {ip 22}`, `ipnettodiaentry` или `iproutedest` и т.д., полезно контролировать при серьезных сбоях и сравнивать их с эталонными значениями. Некоторые переменные важны при анализе эффективности системы, например, `ipfragcreate {ip 19}` или `ipfragfails {ip 18}`, — последняя переменная говорит о том, сколько встретилось пакетов с флагом, запрещающим фрагментацию, в условиях, когда она необходима, что может свидетельствовать о неверном выборе MTU. MTU (Maximum Transmission Unit) используется для определения максимального размера блока (в байтах), который может быть передан на канальном уровне коммуникационного протокола [16].

Рассмотрим средние значения некоторых переменных за сутки. Так, переменные `ip.ipinreceives=21621184`, `ifentry.ifinucastpkts=6235719`, `ifentry.ifinnunicastpkts=1037022` характеризуют средний поток пакетов на входе сетевого интерфейса. Видно, что широковещательные и мультикастинг-пакеты составляют малую долю трафика. Большой поток пакетов типа `nonunicast` обычно говорит о неисправности в сети. Величины `ifentry.ifoutunicastpkts=6246308` и `ifentry.ifoutnunicastpkts=90` характеризуют выходной поток пакетов, соотношение обычных и `nonunicast`-пакетов и здесь нормальное. Сравнимое их значение говорило бы о неисправности сетевого интерфейса данного узла или о некорректной работе сетевого драйвера. Блок переменных `ip.ipinherrors=8`, `icmp.icmpouterrors=0`, `icmp.icmpoutdestunreachs=22` и `ifentry.ifinunknownprotos=2` указывает на число сбоев в сети (если соотнести эти цифры с входным и выходным потоками пакетов, можно сделать вывод о нормальном поведении трафика в данном логическом сегменте на протяжении исследуемого периода). Такие ошибки возможны из-за всплеска шумов или наводок (например, по сети переменного тока или в результате грозы). Определенное беспокойство может вызвать значение `icmpoutdestunreachs`, но это может быть результатом работы программы `ping` или `traceroute` для недоступного узла или опечатка в IP-адресе.

Переменная `icmp.icmpinsrcquenchs=19` весьма важна, так как она отмечает случаи перегрузки. В данном случае таких ситуаций за сутки было мало. Отслеживая эту переменную для разных узлов, можно выявить слабые элементы в сети и скорректировать их параметры (например, увеличить буфер-

ную память). Переменные `tcp.tcpinsegs=148869`, `tcp.tcputsegs=134718` и `tcp.tcpretranssegs=1882` говорят о потоках TCP-пакетов (главного транспортного средства Интернет). Число `tcpretranssegs` характеризует надежность и правильность настройки параметров сети: чем меньше это число, тем лучше. `udp.udpindatagrams=20431650`, `udp.udpoutdatagrams=20319565` указывают на входной и выходной потоки UDP-дейтаграмм. Запись в MIB `udp.udpnoports` является важным диагностическим показателем. Переменные, регистрирующие число тех или иных ошибок и не упомянутые, оказались равными нулю. Количество пакетов `SNMP` в точности совпадает с их числом, посланным и полученным, что говорит об отсутствии какой-либо другой `SNMP`-активности. Контролировать это время от времени также полезно из соображений сетевой безопасности.

Методы предварительной обработки

Предварительная обработка данных осуществляется с помощью вычисления среднего значения ряда и среднеквадратичного отклонения [1, 2, 3] по формулам (9) – (11):

$$\sigma_i^2 \equiv \frac{1}{P-1} \sum_{\alpha=1}^P (x_i^\alpha - \bar{x}_i)^2, \quad (9)$$

$$\bar{x}_i \equiv \frac{1}{P} \sum_{\alpha=1}^P x_i^\alpha, \quad (10)$$

$$\tilde{x}_i = \left(\frac{x_i - \bar{x}_i}{\sigma_i} \right). \quad (11)$$

Для определения выходных нейронов используется интервал $[-1,1]$ и, соответственно, в качестве функции активации для всех нейронов всех слоев будет использоваться гиперболический тангенс:

Экспериментальные данные, полученные по результатам работы подсистемы прогнозирования NeuroNet

Для исследований возможностей разработанной подсистемы прогнозирования NeuroNet были проведены различные эксперименты, основная цель которых – подобрать значения параметров настройки программы, при которых итоговые результаты ее работы содержали наименьшее количество ошибок в прогнозе. Эксперименты проводились на данных, полученных от системных администраторов корпоративной сети RUNNET, которая использовалась как прототип

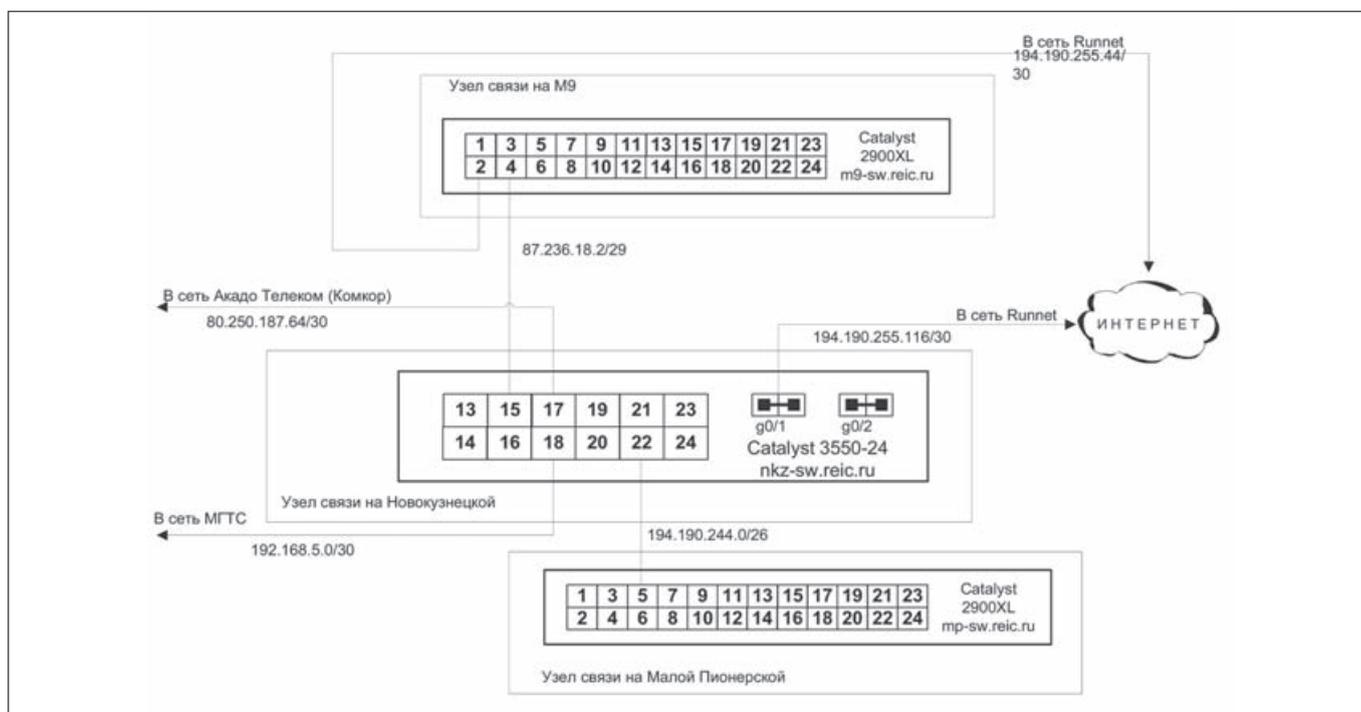


Рис. 3. Структура узла связи корпоративной сети RUNNet

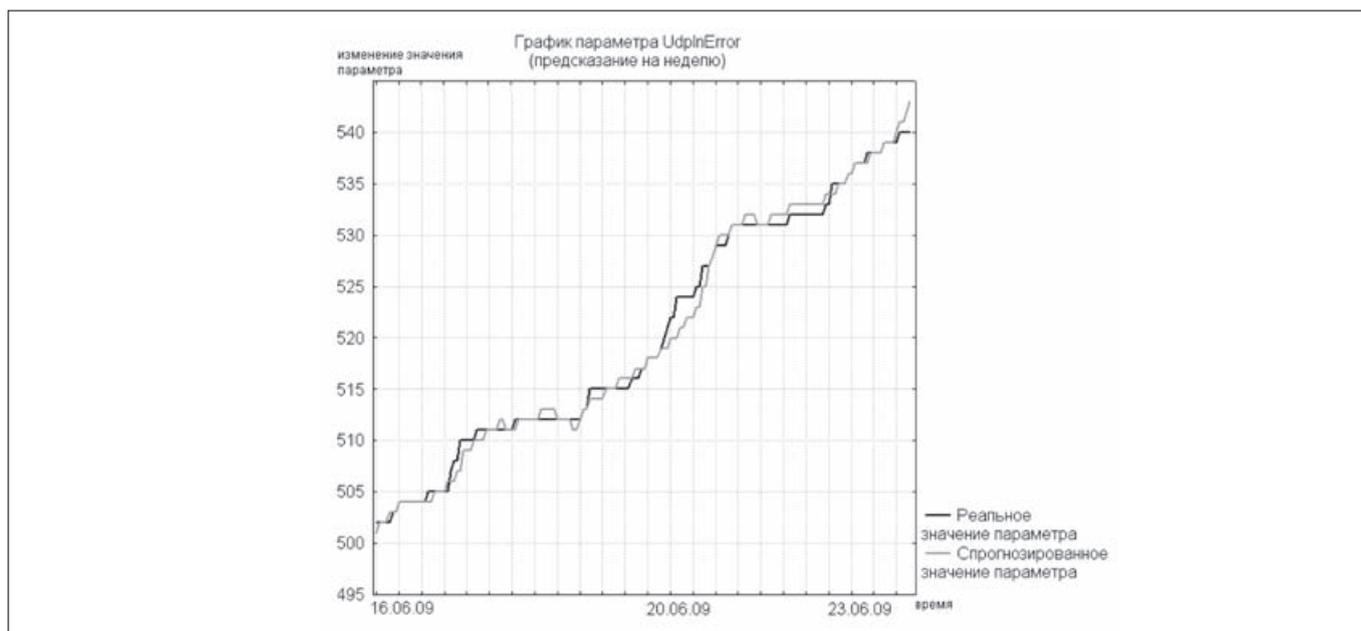


Рис. 4. Реальное и прогнозируемое изменение параметра UdpInErrors с прогнозом изменения параметра на неделю (точность прогноза составляет 90%)

сети ЦУПа. Для построения прогноза статистика собиралась для узла сети RunNet (рис. 3) в течение 2 месяцев, включая критические, с периодом в 1 час.

Проведенные исследования показали, что для эффективного решения поставленной задачи большую роль играет правильная система информативных переменных, а также правильно подобранные выборки, т.к., например, в течение одного месяца

при стабильной работе сервисов нейронная сеть не могла оптимально обучиться, т.к. ей не хватало примеров с критическими значениями параметров.

Для того чтобы иметь возможность оценить, насколько эффективно работает подсистема прогнозирования, в качестве эксперимента ставились опыты по прогнозированию трафика сети по интегральным параметрам, и сравнение итогов с теми,

которые получились в результате опытов прогнозирования по параметрам МІВ устройств. Экспериментальные данные по трафику предоставлялись администраторами сети RUNNet.

Оценивая различные параметры можно увидеть, что анализ трафика по интегральным параметрам дает около 80% правильных предсказаний, в то время как анализ параметров базы данных МІВ – 82-98%, что говорит о целесообразности использования базы данных МІВ для прогнозирования при повышенных требованиях к точности прогноза. Кроме того, исследования показали, что точность прогноза существенно зависит от величины интервала прогнозирования, которую необходимо выбирать также с учетом требований к точности прогнозирования. На рис. 4 представлен пример результатов прогноза.

ЛИТЕРАТУРА

1. Барский А.Б. Логические нейронные сети. – М.: Наука и техника, 2003. 384 с.
2. Боровиков В.П., Ивченко Г.И. Прогнозирование в системе Statistica в среде Windows. – М.: Финансы и статистика, 2006. 368 с.
3. Ежов А.А., Шумский С.А. Нейрокомпьютеринг и его применение в экономике и бизнесе. Серия «Учебники экономико-аналитического института МИФИ» / под ред. проф. Харитонов В.В. – М.: МИФИ, 1998. 224с.
4. Иванов И.А., Леохин Ю.Л. Способ управления компьютерной сетью и система для его реализации // Патент на изобретение № 2309451 от 27.10.2007 – Москва. – Федеральная служба по интеллектуальной собственности, патентам и товарным знакам.
5. Комашинский В.И., Смирнов Д.А. Нейронные сети и их применения в системах управления и связи. – М.: Горячая линия-Телеком, 2003. 94 с.
6. Круглов В.В., Борисов В.В. Искусственные нейронные сети. Теория и практика. – М.: Горячая линия-Телеком, 2002. 382 с.
7. Минаев Ю.Н., Филимонова О.Ю., Бенамеур Л. Методы и алгоритмы идентификации и прогнозирования в условиях неопределенности в нейросетевом логическом базисе. – М.: Горячая Линия – Телеком, 2003. 208 с.
8. Назаров А.В., Лоскутов А.И. Нейросетевые алгоритмы прогнозирования и оптимизации систем. – СПб.: Наука и техника, 2003. 384 с.
9. Нейронные сети. STATISTICA Neural Networks / пер. с англ. – М.: Горячая линия-Телеком, 2000. 182 с.
10. Оссовский С. Нейронные сети для обработки информации / пер. с польского. – М.: Финансы и статистика, 2004. 344 с.
11. Пегат А. Нечеткое моделирование и управление / пер. с англ. – М.: БИНОМ. Лаборатория знаний, 2009. 798 с.
12. Чулюков В.А., Астахова И.Ф., Потапов А.С. и др. Системы искусственного интеллекта. Практический курс: учебное пособие / Под ред. Астаховой И.Ф. – М.: БИНОМ. Лаборатория знаний, 2008. 292 с.
13. <http://www.inlinegroup.ru/projects/3242.php>

Леохин Юрий Львович,
д-р техн. наук, профессор, нач. научно-исследовательской части МГИЭМ.
e-mail: leo@miem.edu.ru