

# Анализ параметров промышленных сетей с применением нейросетевой обработки

Р. Ф. Гибадуллин<sup>1</sup>, Д. В. Лекомцев<sup>1</sup>, М. Ю. Перухин<sup>II</sup>

<sup>I</sup> Казанский национальный исследовательский технический университет им. А.Н. Туполева, г. Казань, Россия

<sup>II</sup> Казанский национальный исследовательский технологический университет, г. Казань, Россия

**Аннотация.** В статье использованы искусственные нейронные сети и диагностическая информация сети для оценки состояния PROFINET (сеть полевого уровня). Особое внимание уделено предварительной обработке данных, которое выполняется с помощью квантования, выравнивания данных, уменьшения количества входов и других методов для создания новых версий набора данных с целью повышения точности. Полученные в ряде экспериментов параметры показывают, какие подходы к предварительной обработке данных дают наилучшие результаты, которые были оценены на двух наборах. Работа открывает возможности для повышения точности обработки данных.

**Ключевые слова:** PROFINET, промышленный Ethernet, диагностика сети, искусственные нейронные сети, машинное обучение, предварительная обработка данных.

DOI 10.14357/20718594200108

## Введение

Анализ рынка промышленных сетей показывает, что Industrial Ethernet занимает 52% рынка [1]. В домашних или офисных сетях гарантия доставки сообщений не всегда важна. Например, если случается потеря информационных пакетов во время видеоконференции, то пользователь увидит только незначительный дефект или даже ничего не заметит. Но в промышленной области очень важно знать, что пакет был доставлен и был доставлен вовремя. Например, потеря пакета с сообщением для запуска машины на производственной линии может остановить всю производственную линию. Поэтому существует потребность в сетях особого типа для промышленных целей, где очень важно знать, что информационные пакеты вовремя отправляются и принимаются.

Таким образом, ненадежная промышленная сеть потенциально является причиной снижения скорости производственных процессов. Важно собирать информацию о состоянии сети и принимать соответствующие меры. Сети могут быть диагностированы с использованием специальных устройств и программного обеспечения. Использование методов машинного обучения может помочь в понимании причин ошибок и дать ключевую информацию для улучшения сетей.

Анализ параметров сетевой инфраструктуры позволяет уменьшить время выявления причины неисправности промышленной сети. В качестве решения предлагается разработать устройство по сбору данных. Для получения каких-либо правил по этим данным требуется высокая квалификация от эксперта. Поэтому для работы были использованы нейросети, обученные с учителем, так как имеется большое количество

✉ Гибадуллин Руслан Фаршатович. E-mail: landwatersun@mail.ru

записей (данные с метками), в которых есть параметр *networkcondition*. Обучение нейросети довольно ресурсоемкая задача, однако выбранная ее модель является легкой для вычисления даже на диагностическом устройстве. Это является причиной выбора именно нейросетевого подхода к решению данной задачи.

Искусственные нейронные сети (ИНС) – универсальные инструменты для решения прикладных задач. Они могут использоваться для поиска шаблонов, прогнозирования состояния сети и т.д. Эта работа нацелена на использование ИНС для определения состояний сети в ходе ее неисправности. Тем самым, использование ИНС помогает повысить общую надежность сети.

В настоящее время наиболее широко используются сети на основе Ethernet [1]. Но обычная реализация Ethernet не подходит для промышленных целей. Поэтому Ethernet имеет промышленные реализации, такие как PROFINET, EtherNet/IP, EtherCAT, SERCOS III и POWERLINK [2]. Эти реализации имеют свои преимущества из-за ограниченной сферы их применения.

PROFINET (Process Field Network – сеть полевого уровня) – открытый промышленный стандарт [3, с.2171-2180]. Он основан на промышленном Ethernet, предназначен для промышленных сетей и часто используется для подключения сети PROFIBUS к обычной сети Ethernet. PROFINET предоставляет возможность организовать сети с несколькими master-устройствами. Более того, поскольку PROFINET основан на технологии Ethernet, он может быть интегрирован в существующие офисные (Ethernet) сети.

В работе [4] предложен метод для обнаружения аномалий в PROFINET, используя нейронные сети. Данный метод распознает четыре состояния сети. В работе [5, с.658-669] использован Apache Hadoop (свободно распространяемый набор утилит, библиотек и фреймворк для разработки и выполнения распределенных программ), чтобы предупредить сбой сетевого оборудования. В [5] использовано большое количество данных (4 миллиона записей каждый день в течение двух месяцев). Полученная точность предсказания аномалий – 99.9%. В работе [6, с.18553-18565] предложен метод прогнозирования отказа оптического оборудования в оптических сетях на основе метода опорных векторов и двойного экспоненциального сглаживания. В [6] точность предсказания отказа оптического оборудования

составила 95%. В работе [7, с.242-247] предложен метод диагностики сетей на базе PROFIBUS. Авторы данной работы использовали информацию с физического, канального и пользовательского уровней.

В статье оценивается состояние сети с использованием данных, полученных с анализатором сетей PROFINET-INspector® NT [8]. Новизна работы заключается в использовании перспективных методов предварительной обработки данных.

## 1. Постановка задачи и предпосылки к ее решению

Суть работы, описанной в данной статье, заключается в разработке программного обеспечения для оценки качества промышленной сети. Для оценки качества сети используется PROFINET-INspector® NT (Рис. 1), который выполняет сетевые измерения. Ниже перечислены измеряемые параметры и их описание.

*Lostnodes* показывает, сколько раз устройство выходило из строя. Если оно не отвечает в течение некоторого времени и соответствует таймауту, появляется сбой узла.

*Highpriorityalarm, lowpriorityalarms* сообщают количество тревог. PROFINET поддерживает тревоги, которые могут иметь высокий или низкий приоритет. Они могут быть получены с разных устройств и в разных условиях, например, при высокой температуре, обрыве провода и т.д.

*Errortelegrams* показывает количество ошибочных или уничтоженных телеграмм в сети.



Рис. 1. Устройство диагностики PROFINET-INspector® NT

*Framebreach* – логический параметр сообщает, что устройство не отправило телеграмму. Также он означает, что существует проблема дрожания.

*Frameovertake* показывает, что новая телеграмма была отправлена до того, как была отправлена более старая. Это может означать, что сетевая нагрузка слишком высока или устройство работает неправильно.

*Loadratio* – количество трафика PROFINET в той же сети. Рекомендуется поддерживать это соотношение нагрузки не ниже 1:100.

*Jitter* показывает задержку телеграммы. В промышленных сетях джиттер до 50% считается приемлемым. Более высокие значения показывают проблемы в сети.

*Broadmulticastframes* – количество телеграмм, которые должны быть получены всеми членами сети, и количество телеграмм, отправленных группе получателей. Многоадресные телеграммы значительно увеличивают нагрузку на сеть и должны использоваться редко.

*Temperature* показывает температуру измерительного устройства.

*Cycletimemaximum*, *cycletimeminimum* – сообщают время, затрачиваемое на отправку сообщения и получение ответов. Значения этих параметров должны оставаться почти постоянными.

*Utilizationmaximum*, *utilizationaverage* показывают среднюю и максимальную нагрузки на сеть в течение одной минуты в процентах соответственно. Максимальная сетевая нагрузка 100% равна максимальной сетевой нагрузке линии со скоростью 100 Мбит/с. Рекомендуется поддерживать сетевую нагрузку ниже 20%.

*Restartnodes* – количество перезапусков узлов в сети.

*Unavailabilitycounter* – сообщает сколько раз подряд устройство недоступно. Когда устройство снова будет доступно, параметр сбрасывается в 0.

*Networkcondition* – показывает рассчитанное состояние сети, основанное на многих параметрах. Максимальное значение 100 указывает на наилучшее состояние сети, тогда как 0 используется для обозначения наихудшего.

Кроме того, база данных содержит следующую информацию: параметр *id* предназначен для внутренней обработки данных (первичный ключ), *scanstart* – для хранения

даты и времени измерения; параметр *affecteddevicepart\_id* показывает, какое устройство было измерено.

Для неподготовленного пользователя данные параметры не являются информативными, и сделать вывод о качестве сети не представляется возможным. Однако эти параметры могут быть полезны для автоматической оценки качества сети.

Задачу по разработке ИНС можно разделить на две части: предварительная обработка данных и обучение ИНС; проверка и тестирование (Рис. 2).

В первой части все данные используются в качестве входных данных, а параметр *networkcondition* является выходным. *id*, *scanstart* и *affecteddevicepart\_id* могут не использоваться, поскольку они не могут предоставить информацию о сети, отличной от тестовой. Во второй части анализируются зашумленные данные, и готовится тренировочный набор данных. Предварительная обработка данных состоит из нескольких этапов. Требуется проанализировать входные данные, чтобы решить, как подготовить данные наилучшим образом. Улучшение обучения нейронной сети достигается отклонением значений после десятичной точки или принятием значений как равных другим близким значениям (т.е. использованием квантования).

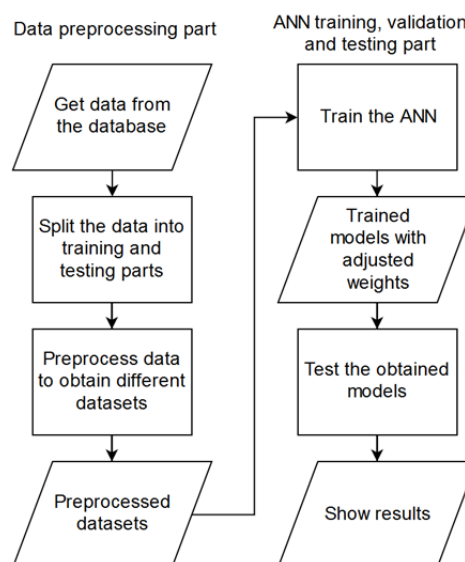


Рис. 2. Этапы построения ИНС

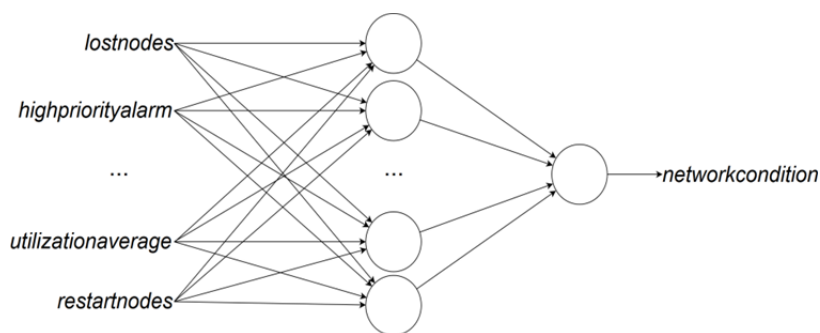


Рис. 3. Используемая модель ИНС

Данные разделяются на следующие типы:

1) Входные данные включают два набора. Первый состоит из измерений, выполненных в период с 07.2016 по 10.2016 и содержит 109170 записей, второй – из измерений с 01.2017 по 04.2017 и содержит 131270 записей. Оба набора представляют данные с одного и того же сетевого устройства.

2) Данные валидации. Чтобы избежать переобучения, была использована валидация после каждой эпохи во время обучения. 50% собранных записей используются для обучения, а другая половина – для проверки.

3) Тестовые данные. Чтобы оценить результат, необходим тестовый набор данных параметра *networkcondition*. Этот параметр напрямую указывает на сетевые проблемы, поэтому хорошо подходит для оценки состояния сети. Для получения более точного результата используются разные методы предварительной обработки данных. Вход ИНС содержит 16 входов. Следующий полносвязный слой состоит из 16 нейронов и функции активации ReLU. Выходной слой состоит из одного нейрона и имеет сигмоидальную функцию активации. Этот слой представляет параметр *networkcondition*. Количество входов зависит от входных данных и автоматически настраивается.

Модель ИНС показана на Рис.3.

## 2. Предварительная обработка данных

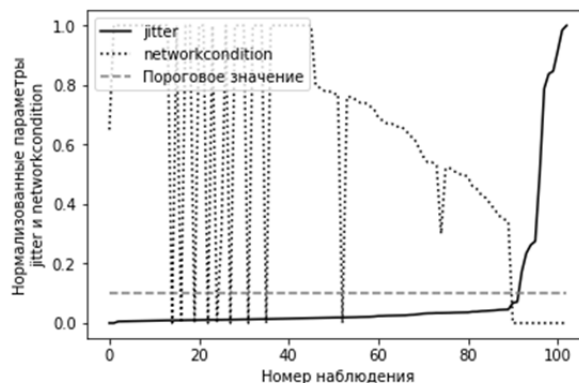
Предварительная обработка данных состоит из нескольких этапов:

1) Выравнивание данных. Для этой операции удаляются случайные строки из преобладающих множеств, чтобы получить более или менее равномерное распределение.

2) Обрезка по пороговому значению. Некоторые данные могут быть усечены по пороговому

значению в тех случаях, когда при увеличении рассматриваемого значения выходной сигнал не изменяется. Этот подход был применен к параметрам *jitter*, *unavailabilitycounter*, *restartnodes* и *framebreach*. Для графического представления данных были созданы две диаграммы. Данные сортируются в порядке возрастания параметра *jitter*. Рис. 4 содержит как нормированные значения *jitter*, так и *networkcondition*, чтобы показать поведение *networkcondition* по отношению к *jitter*. На Рис. 5 изображены данные после обрезки по пороговому значению. Данные были нормализованы, чтобы показать оба параметра на одном графике, сохраняя пропорции.

3) Квантование данных. Квантование было использовано для того, чтобы рассматривать параметры близкие по значению как дубликаты. Пример квантования параметра *jitter* изображен на Рис. 6. Полученные значения параметра *jitter* имеют меньше уникальных значений. Пустые значения заменены на -1, так как ИНС позволяют работать только с числами. Для масштабирования данных была использована min-max-нормализация. Все нормированные значения находятся в диапазоне от 0 до 1 включительно:

Рис. 4. Параметр *jitter* до обрезки по пороговому значению

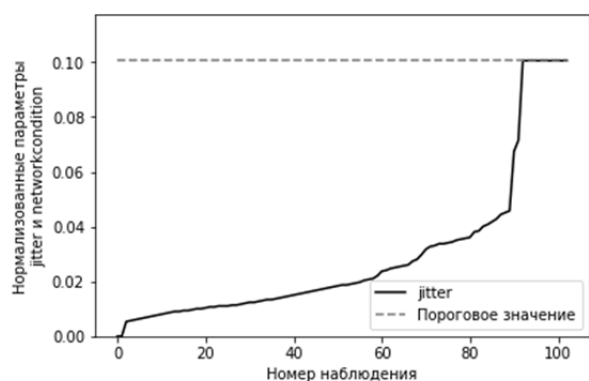


Рис. 5. Результат обрезки по пороговому значению

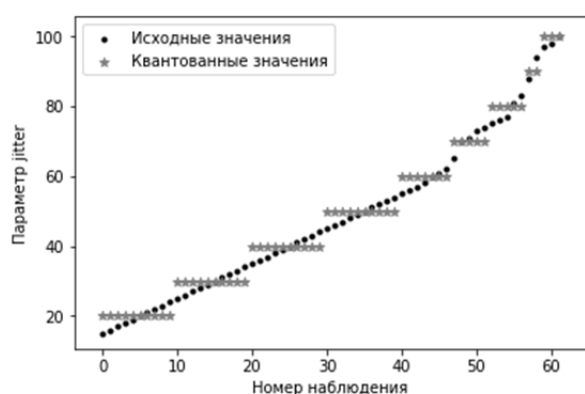


Рис. 6. Использование квантования для параметра jitter

$$x_{norm,i} = \frac{x_i - \min(X)}{\max(X) - \min(X)},$$

где  $x_{norm,i}$  – нормализованное значение  $x_i$ ,  $\forall x_i \in X, i = 1, 2, \dots, N$ .

Данные нормализуются до разделения на тренировочный и тестовый наборы данных.

4) Организация двоичного выхода. Можно рассматривать выходные данные ИНС как двоично размеченные данные. Выходной сигнал равен нулю, если сеть имеет проблемы, и единице, когда сеть работает нормально.

5) Удаление входных данных. Параметры *unavailabilitycounter*, *jitter*, *framebreach*, *lostinodes* и *cycletimemaximum* использованы для предсказания параметра *networkcondition*.

6) Кодирование групп элементов. Выделение групп разных значений характеризует следующий пример. Параметр *cycletimemaximum* содержит несколько значений в окрестности 2000; параметр *networkcondition* в этой точке меньше или равен 30. Существует 4 состояния

Табл. 1. Кодирование значений *cycletimemaximum*

Номер состояния	Условие	Результат
1	<i>cycletimemaximum</i> = -1	0
2	$-1 < cycletimemaximum < 99999$	1/3
3	$10000 < cycletimemaximum < 1000000$	2/3
4	<i>cycletimemaximum</i> > 1000000	1

*cycletimemaximum*: -1 (пустое значение), около 2000, около 130000 и более 1000000. Было решено закодировать эти группы по условию, которое показано в Табл. 1.

### 3. Обучение ИНС

Для работы с ИНС был выбран высокоуровневый API для создания моделей глубокого обучения Keras [9] с использованием библиотеки численного вычисления в Python – Theano [10, с.3-10].

Тренировочные наборы разделены на две части: учебные и валидационные. Набор данных для валидации составляет 50% от тренировочного. В работе использован набор валидаций для сбора данных в ходе учебного процесса. При этом отсутствует переобучение и использование техники ранней остановки. Точность может колебаться от эпохи до эпохи. В качестве параметра ранней остановки была использована ошибка валидации (обобщения). Ранняя остановка срабатывает в том случае, если в течение 1000 эпох ошибка обобщения не улучшается.

В ходе экспериментов установлено, что использование входных параметров для обучения ИНС улучшило точность обнаружения безошибочного состояния сети до 87.70%. Точность обнаружения сетевых проблем составила 98.91%. После удаления всех дубликатов из обучающего набора получена более низкая точность предсказания равная 85.94%.

Более подробная информация по результатам обучения показана в Табл. 2, где:

$N_2$  – номер эксперимента.

Квантование – данные предобработаны с использованием квантования.

Уменьшенное количество входов – входы ИНС содержат только параметры *unavailabilitycounter*, *jitter*, *framebreach*, *lostinodes*, *cycletimemaximum*.

Табл. 2. Результаты исследования

№	Квантование	Уменьш. кол-во входов	Двоичн. выход	Дубликаты удалены	Данные выровнены	Кол-во векторов обучения	Кол-во тестовых векторов с пробл.	Кол-во тестовых векторов	Точность предсказания неисправности в сети	Точность предсказания безошибочного состояния сети
1	-	-	-	-	-	109170	542	130728	16.05	100
2	-	-	-	-	+	321	542	130728	16.24	99.94
3	-	+	-	+	+	587	517	27477	17.02	99.98
4	+	-	-	-	+	140	459	2065	19.17	98.35
5	+	-	+	-	+	140	459	2065	98.91	87.55
6	+	+	+	-	+	140	459	2065	98.91	87.70
7	+	+	+	+	+	22	64	15	85.94	80
8	+	+	-	-	+	140	91	15	43.96	100

*Двоичный выход* – параметр *networkcondition* представлен таким образом, что выход ИНС может быть только 0 или 1.

*Дубликаты удалены* – набор данных для обучения, состоит только из уникальных векторов обучения.

*Данные выровнены* – данные в обучающем наборе не содержат преобладающих выходных параметров.

*Количество векторов обучения* показывает количество обучающих векторов в обучающем наборе.

*Количество тестовых векторов с проблемами* показывает количество тестовых векторов в сети с проблемами (*networkcondition* < 100).

*Количество тестовых векторов* показывает количество тестовых векторов в нормально работающей сети (*networkcondition* = 100).

*Точность в сети с проблемами* – точность, полученная в сети с проблемами (*networkcondition* < 100).

*Точность в сети без проблем* – точность, полученная в нормально работающей сети (*networkcondition* = 100).

Чтобы улучшить результат, целесообразно использовать соответствующий метод для выбора параметров входных данных. Но главное здесь – сами данные. Текущие данные неоднозначны. Поэтому еще один возможный подход к повышению точности заключается в использовании данных временного ряда.

После квантования и удаления дубликатов большое количество данных было удалено. Следует, по возможности, использовать больше данных для обучения ИНС.

## Заключение

По итогам работы реализована ИНС для обнаружения проблем в сети с использованием обучения с учителем и собранных диагностических данных. Данные обучения и тестирования были предварительно обработаны с использованием предложенных методов. Полученные наборы данных были использованы для проведения ряда экспериментов. Полученные данные позволяют утверждать, что ИНС способна выполнить оценку состояния сети. Наилучшая точность обнаружения сетевых проблем составляет 98.91%, та же самая ИНС оценивает хорошо работающую сеть с точностью 87.70%. В некоторых случаях мы получили крайне малый размер тренировочного набора, но, тем не менее, он обладает хорошими свойствами обобщения.

Для получения более убедительных выводов по анализу параметров промышленных сетей ставятся задачи по разрешению следующих вопросов:

- Какая модель ИНС лучше подходит для улучшения точности оценки сети?
- Какие данные действительно необходимы для обучения?
- Как найти правильное пороговое значение для предварительной обработки данных?
- Какой шаг квантования использовать для каждого параметра?
- Как выполнить выравнивание данных без потери данных, необходимых для обучения?
- Стоит ли рассматривать данные в качестве временного ряда?

- Достаточно ли данных для обучения?
- Можно ли предугадать проблемы в сети?
- Можно ли использовать обучение без учителя?

## Литература

1. What is the difference between Ethernet and Industrial Ethernet [Электронный ресурс] URL: <http://www.innovasic.com/news/industrial-ethernet/what-is-the-difference-between-ethernet-and-industrial-ethernet/> (дата обращения: 19.04 2019).
2. The Industrial Ethernet Book [Электронный ресурс] URL: <http://www.iebmedia.com/index.php?id=12993&parentid=74&showdetail=true> (дата обращения: 19.04 2019).
3. Sestito G., Turcato A., Dias A., Rocha M., Silva M., Ferrari P., Brandao D. A Method for Anomalies Detection in Real-Time Ethernet Data Traffic Applied to PROFINET, IEEE Transactions on Industrial Informatics, vol. 14, no. 4, 2018, pp. 2171-2180.
4. Shuan L., Fei T., King S., Xiaoning G., Mein L. Network Equipment Failure Prediction with Big Data Analytics, International Journal of Advances in Soft Computing & Its Applications, vol. 8, no. 3, 2016.
5. Mossin A., Brandao D., Sestito G., Torres R. Automatic Diagnosis for Profibus Networks, Journal of Control, Automation and Electrical Systems, vol. 27, no. 6, 2016, pp. 658-669.
6. Wang Z., Zhang M., Wang D., Song C., Liu M., Li J., Lou L., Liu Z. Failure prediction using machine learning and time series in optical network, Opt. Express, vol. 25, no. 16, 2017, pp. 18553-18565.
7. Sestito G., Mossin E., Brandão D., Dias A. Artificial neural networks and signal clipping for Profibus DP diagnostics, 12th IEEE International Conference on Industrial Informatics (INDIN), 2014. IEEE, 2014, pp. 242-247.
8. PROFINET Diagnosis – PROFINET-INSpektor® NT - Indu-Sol [Электронный ресурс] URL: <https://www.indu-sol.com/en/products/profinet/diagnostics/profinet-inspektorr-nt/> (дата обращения: 19.04 2019).
9. Chollet, F. (2015) keras, GitHub [Электронный ресурс] URL: <https://github.com/fchollet/keras> (дата обращения: 19.04 2019).
10. Bergstra J., Breuleux O., Bastien F., Lamblin P., Pascanu R., Desjardins G., Turian J., Warde-Farley D., Bengio Y. Theano: a CPU and GPU Math Expression Compiler in Python в Proceedings of the Python for Scientific Computing Conference (SciPy). IEEE, 2010, pp. 3-10.

## Neural Network Data Processing for Analysis of the Industrial Networks Parameters

R. F. Gibadullin<sup>†</sup>, D. V. Lekomtsev<sup>†</sup>, M. Y. Perukhin<sup>‡</sup>

<sup>†</sup> Kazan National Research Technical University named after A. N. Tupolev, Kazan, Russia

<sup>‡</sup> Kazan National Research Technological University, Kazan, Russia

**Abstract.** We used artificial neural networks and diagnostic network information to assess the condition of PROFINET (Process Field Network). An artificial neural network determines whether the network works fine or not. An important part of this work is data preprocessing. An essential part of the work is data preprocessing. It is done using quantization, data aligning, reducing the number of inputs and other preprocessing techniques to create a new version of the dataset to improve the accuracy. The obtained data makes possible to do a number of experiments and to find out what approach of data preprocessing shows the best results. The results were evaluated on two datasets. The first dataset contains diagnostic data of a well-functioning network, and the second one consists of data in which network problems were detected. The highest accuracy obtained in this work is 98.91% of recognizing problems in the network and the accuracy of 87.70% when the network is working fine. The work also opens up opportunities to improve accuracy in the future.

**Keywords:** PROFINET, Industrial Ethernet, Network Diagnostics, Artificial Neural Networks, Machine Learning, Data Preprocessing.

DOI 10.14357/20718594200108

## References

1. What is the difference between Ethernet and Industrial Ethernet. Available at: <http://www.innovasic.com/news/industrial-ethernet/what-is-the-difference-between-ethernet-and-industrial-ethernet/> (accessed April 19, 2019).
2. The Industrial Ethernet Book. Available at: <http://www.iebmedia.com/index.php?id=12993&parentid=74&showdetail=true> (accessed April 19, 2019).
3. Sestito, G., A. Turcato, A. Dias, M. Rocha, M. Silva, P. Ferrari, and D. Brandao. 2018. A Method for Anomalies Detection in Real-Time Ethernet Data Traffic Applied to PROFINET [Transactions on Industrial Informatics]. 14(4):2171-2180. doi: 10.1109/TII.2017.2772082
4. Shuan, L., T. Fei, S. King, G. Xiaoning, and L. Mein. 2016. Network Equipment Failure Prediction with Big Data Analytics [International Journal of Advances in Soft Computing & Its Applications]. 8(3).
5. Mossin, A., D. Brandao, G. Sestito, and R. Torres. 2016. Automatic Diagnosis for Profibus Networks [Journal of Control, Automation and Electrical Systems]. 27(6):658-669.
6. Wang, Z., M. Zhang, D. Wang, C. Song, M. Liu, J. Li, L. Lou, and Z. Liu. 2017. Failure prediction using machine learning and time series in optical network [Opt. Express]. 25(16):18553-18565.
7. Sestito, G., E. Mossin, D. Brandão, and A. Dias. 2014. Artificial neural networks and signal clipping for Profibus DP diagnostics [12th IEEE International Conference on Industrial Informatics (INDIN)]. Brazil. 242-247. doi: 10.1109/INDIN.2014.6945515
8. PROFINET Diagnosis – PROFINET-INSpektor® NT – Indu-Sol. Available at: <https://www.indu-sol.com/en/products/profinet/diagnostics/profinet-inspektorr-nt/> (accessed April 19, 2019).
9. Chollet, F. (2015) keras, GitHub. Available at: <https://github.com/fchollet/keras> (accessed April 19, 2019).
10. Bergstra, J., O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio. 2010. Theano: a CPU and GPU Math Expression Compiler in Python [Proceedings of the Python for Scientific Computing Conference (SciPy)], Austin. 3-10.