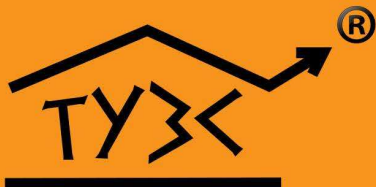


ISSN: 1813-324X (print)
2712-8830 (online)



Том 9. № 6
2023

ТРУДЫ УЧЕБНЫХ ЗАВЕДЕНИЙ СВЯЗИ

Темы номера:

- ✓ Электромагнитная доступность абонентских терминалов Starlink
- ✓ Машинное обучение vs поиск уязвимостей
- ✓ Границы оценивания параметров сигналов

Vol. 9. Iss. 6
2023

PROCEEDINGS
OF TELECOMMUNICATION UNIVERSITIES

Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Санкт-Петербургский государственный университет телекоммуникаций
им. проф. М.А. Бонч-Бруевича» (СПбГУТ)

Научный журнал

**ТРУДЫ
УЧЕБНЫХ ЗАВЕДЕНИЙ СВЯЗИ**

Том 9. № 6

Proceedings of Telecommunication Universities

Vol. 9. Iss. 6

Санкт-Петербург

2023

Описание журнала

Научный журнал. Включен в Перечень рецензируемых научных изданий, в которых должны быть опубликованы основные научные результаты диссертаций на соискание ученой степени кандидата наук, на соискание ученой степени доктора наук (распоряжение Минобрнауки России № 21-р от 12.02.2019), по специальностям (распоряжение № 33-р от 01.02.2022):

1.2.2. Математическое моделирование, численные методы и комплексы программ

2.2.6. Оптические и оптико-электронные приборы и комплексы

2.2.13. Радиотехника, в том числе системы и устройства телевидения

2.2.14. Антенны, СВЧ-устройства и их технологии

2.2.15. Системы, сети и устройства телекоммуникаций

2.2.16. Радиолокация и радионавигация

2.3.1. Системный анализ, управление и обработка информации

2.3.6. Методы и системы защиты информации, информационная безопасность

Выпускается с 1960 года. Выходит 6 раз в год. Издается на русском и английском языках.

Редакционный совет

Киричек Р.В. <i>Главный редактор</i>	д.т.н., доцент, Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича (СПбГУТ), г. Санкт-Петербург, Россия
Владыко А.Г. <i>Зам. Главного редактора</i>	к.т.н., доцент, Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича (СПбГУТ), г. Санкт-Петербург, Россия
Буйневич М.В. <i>Шеф-редактор</i>	д.т.н., проф., Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича (СПбГУТ), г. Санкт-Петербург, Россия
Зеневич А.О.	д.т.н., проф., Белорусская государственная академия связи, г. Минск, Республика Беларусь
Розанов Н.Н.	д.ф.-м.н., проф., чл.-корр. РАН, АО «Государственный оптический институт им. С.И. Вавилова» (ГОИ), г. Санкт-Петербург, Россия
Дукельский К.В.	д.т.н., доцент, АО «Государственный оптический институт им. С.И. Вавилова» (ГОИ), г. Санкт-Петербург, Россия
Кучерявый Е.	PhD, Технологический университет Тампере, г. Тампере, Финляндия
Каримов Б.Т.	к.т.н., доцент, Институт электроники и телекоммуникаций, Кыргызский государственный технический университет И. Раззакова (КГТУ), г. Бишкек, Кыргызстан
Тиамийу О.А.	PhD, Университет Илорина, г. Илорин, Нигерия
Козин И.Д.	д.ф.-м.н., проф., Алматинский университет энергетики и связи, г. Алма-Аты, Казахстан
Самуйлов К.Е.	д.т.н., проф., Российский университет дружбы народов (РУДН), г. Москва, Россия
Степанов С.Н.	д.т.н., проф., Московский технический университет связи и информатики (МТУСИ), г. Москва, Россия
Росляков А.В.	д.т.н., проф., Поволжский государственный университет телекоммуникаций и информатики (ПГУТИ), г. Самара, Россия
Кучерявый А.Е.	д.т.н., проф., Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича (СПбГУТ), г. Санкт-Петербург, Россия
Канаев А.К.	д.т.н., проф., Петербургский университет путей сообщения имени Александра I (ПГУПС), г. Санкт-Петербург, Россия
Новиков С.Н.	д.т.н., проф., Сибирский государственный университет телекоммуникаций и информатики (СибГУТИ), г. Новосибирск, Россия
Дворников С.В.	д.т.н., проф., Военная академия связи им. Маршала Советского Союза С.М. Буденного (ВАС), г. Санкт-Петербург, Россия
Коржик В.И.	д.т.н., проф., Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича (СПбГУТ), г. Санкт-Петербург, Россия
Ковалгин Ю.А.	д.т.н., проф., Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича (СПбГУТ), г. Санкт-Петербург, Россия

Регистрационная информация

Журнал зарегистрирован Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций: ПИ № 77-77501 от 17.01.2020 г. (пред. рег. № 77-17986 от 07.04.2004 г.)

Подписной индекс в объединенном каталоге «ПРЕССА РОССИИ»: 59983

Размещение в РИНЦ (elibrary.ru) по договору: № 59-02/2013R от 20.02.2013

Контактная информация

Учредитель и издатель: Федеральное государственное бюджетное образовательное учреждение высшего образования «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича» (СПбГУТ)

Адрес учредителя: 191186, Санкт-Петербург, набережная реки Мойки, д. 61, литера А

Адрес редакции: 193232, Санкт-Петербург, пр. Большевиков, 22/1, к. 334/2
Тел.: +7 (812) 326-31-63, м. т. 2022, +79643759970
E-mail: tuzs@sut.ru
Web: <http://tuzs.sut.ru>
ВК: <http://vk.com/spbtuzs>

Description

Scientific journal. The journal is included in the List of reviewed scientific publications, in which the main scientific results of dissertations for the degree of candidate of science and for the degree of doctor of science should be published (order of the Ministry of Education and Science of Russia No 21-r of 12 February 2019) in the field of (order of the Ministry of Education and Science of Russia No 33-r of 01 February 2022):

1.2.2. Mathematical modeling, numerical methods and complexes of programs

2.2.6. Optical and optoelectronic devices and complexes

2.2.13. Radio engineering, including television systems and devices

2.2.14. Antennas, microwave devices and its technologies

2.2.15. Systems, networks and telecommunication devices

2.2.16. Radiolocation and radio navigation

2.3.1. System analysis, management and information processing

2.3.6. Methods and systems of information security, cybersecurity

Since 1960. Published 6 times per year. Published in Russian and English.

Editorial Board

R.V. Kirichek <i>Editor-in-chief</i>	DSc, associate prof., The Bonch-Bruевич Saint-Petersburg State University of Telecommunications (SPbSUT), Saint-Petersburg, Russia
A.G. Vladiko <i>Deputy editor-in-chief</i>	PhD, associate prof., The Bonch-Bruевич Saint-Petersburg State University of Telecommunications (SPbSUT), Saint-Petersburg, Russia
M.V. Buinevich <i>Chief editor</i>	DSc, prof., The Bonch-Bruевич Saint-Petersburg State University of Telecommunications (SPbSUT), Saint-Petersburg, Russia
A.O. Zenevich	DSc, prof., Belarusian State Academy of Communications, Minsk, Republic of Belarus
N.N. Rozanov	DSc, prof., member-corr. RAS, Open Joint Stock Company «S.I. Vavilov State Optical Institute» (SOI), Saint-Petersburg, Russia
K.V. Dukel'skii	DSc, associate prof., Open Joint Stock Company «S.I. Vavilov State Optical Institute» (SOI), Saint-Petersburg, Russia
Y. Koucheryav	PhD, Tampere University of Technology, Tampere, Finland
B.T. Karimov	PhD, Institute of Electronics and Telecommunications, Kyrgyz State Technical University named after I. Razzakov, Bishkek, Kyrgyzstan
O.A. Tiarniyu	PhD, University of Ilorin, Ilorin, Nigeria
I.D. Kozin	DSc, prof., Almaty University of Power Engineering and Telecommunications, Almaty, Kazakhstan
K.E. Samuilov	DSc, prof., Peoples' Friendship University (RUDN), Moscow, Russia
S.N. Stepanov	DSc, prof., Moscow Technical University of Communication and Informatics (MTUCI), Moscow, Russia
A.V. Roslyakov	DSc, prof., Povolzhskiy State University of Telecommunications and Informatics (PSUTI), Samara, Russia
A.E. Koucheryav	DSc, prof., The Bonch-Bruевич Saint-Petersburg State University of Telecommunication (SPbSUT), Saint-Petersburg, Russia
A.K. Kanaev	DSc, prof., Emperor Alexander I-st Petersburg State Transport University (PSTU), Saint-Petersburg, Russia
S.N. Novikov	DSc, prof., Siberian State University of Telecommunications and Information Sciences (SibSUTIS), Novosibirsk, Russia
S.V. Dvornikov	DSc, prof., Military Academy of Telecommunications named after Marshal Union S.M. Budyonny, Saint-Petersburg, Russia
V.I. Korzhik	DSc, prof., The Bonch-Bruевич Saint-Petersburg State University of Telecommunication (SPbSUT), Saint-Petersburg, Russia
Yu.A. Kovalgin	DSc, prof., The Bonch-Bruевич Saint-Petersburg State University of Telecommunication (SPbSUT), Saint-Petersburg, Russia

Registration Information

Registered by Federal Service for Supervision of Communications, Information Technology and Mass Media on 17.01.2020: PI No. 77-77501 (prev. reg. on 04.07.2004: No. 77-17986)

Subscription index for joint catalog «PRESSA ROSSII»: 59983

Accommodation in RINC (elibrary.ru) by agreement on 20.02.2013: No. 59-02/2013R

Contact Information

Publisher:	Federal State Budget-Financed Educational Institution of Higher Education «The Bonch-Bruевич Saint-Petersburg State University of Telecommunications» (SPbSUT)	Post address:	193232, Saint Petersburg, Prospekt Bolshevikov, 22/1
Publisher address:	191186, Saint Petersburg, Moika river embankment, 61-A	Phone:	+7 (812) 326-31-63, local 2022, +79643759970
		E-mail:	tuzs@sut.ru
		Web:	http://tuzs.sut.ru

СОДЕРЖАНИЕ

CONTENTS

КОМПЬЮТЕРНЫЕ НАУКИ И ИНФОРМАТИКА

- | | | |
|---|---|--|
| <p>Попов О.В., Тумашов А.В.,
Борисов Г.Н., Коровин К.О.
Математическая модель несимметричного вибратора
с вынесенной точкой питания.
Часть 3. Коэффициент усиления</p> | 6 | <p>Popov O., Tumashov A.,
Borisov G., Korovin K.
Mathematical model of the unbalanced monopole
feed. Part 3. Gain</p> |
|---|---|--|

ЭЛЕКТРОНИКА, ФОТОНИКА, ПРИБОРОСТРОЕНИЕ И СВЯЗЬ

- | | | |
|--|----|---|
| <p>Брусин Е.А.
Границы оценивания параметров сигналов с прямым
расширением спектра</p> | 25 | <p>Brusin E.
Direct sequence spread spectrum signal's
parameters estimation bounds</p> |
| <p>Дворников С.В.
Электромагнитная доступность
абонентских терминалов Starlink</p> | 34 | <p>Dvornikov S.
Electromagnetic accessibility
of Starlink user terminals</p> |
| <p>Дмитриева Ю.С., Окунева Д.В., Елагин В.С.
Анализ методов идентификационного трафика
для управления ресурсами в SDN</p> | 42 | <p>Dmitrieva J., Okuneva D., Elagin V.
Analyzing traffic identification methods
for resource management in SDN</p> |

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ТЕЛЕКОММУНИКАЦИИ

- | | | |
|---|----|---|
| <p>Васинев Д.А., Соловьев М.В.
Предложения по построению
универсального фаззера протоколов</p> | 59 | <p>Vasinev D., Solovlev M.
Proposals for universal protocol fuzzers
construction</p> |
| <p>Израилов К.Е.
Методология реверс-инжиниринга машинного кода.
Часть 2. Статическое исследование</p> | 68 | <p>Izrailov K.
Methodology for machine code reverse engineering.
Part 2. Static investigation</p> |
| <p>Леонов Н.В., Буйневич М.В.
Машинное обучение vs поиск уязвимостей в
программном обеспечении: анализ применимости
и синтез концептуальной системы</p> | 83 | <p>Leonov N., Buinevich M.
Machine learning vs software vulnerability
detection: applicability analysis and conceptual
system synthesis</p> |
| <p>Неровный В.В., Коратаев П.Д.,
Облов П.С., Толстых М.Ю.
Характеристики уязвимости аппаратуры
потребителей глобальных навигационных
спутниковых систем к спуфинг-атакам</p> | 95 | <p>Nerovny V., Korataev P.,
Oblov P., Tolstykh M.
Vulnerability characteristics of global navigation
satellite system consumer equipment
to spoofing attacks</p> |

РЕЗУЛЬТАТЫ ИССЛЕДОВАНИЯ МОЛОДЫХ УЧЕНЫХ

- | | | |
|--|-----|---|
| <p>Курта П.А.
Эффективная модель интерфейса
взаимодействия пользователя с информационным
сервисом запросного типа</p> | 102 | <p>Kurta P.
An Efficient Interface Model of User Interaction
with a Query-Type Information Service</p> |
|--|-----|---|

КОМПЬЮТЕРНЫЕ НАУКИ И ИНФОРМАТИКА

1.2.2 – Математическое моделирование, численные методы и комплексы программ

Научная статья

УДК 621.3.011.1

DOI:10.31854/1813-324X-2023-9-6-6-23



Математическая модель несимметричного вибратора с вынесенной точкой питания.

Часть 3. Коэффициент усиления

- Олег Вениаминович Попов¹✉, ov.popov@mail.ru
Андрей Витальевич Тумашов¹, ice47reg@yandex.ru
Георгий Николаевич Борисов¹, georgiiborisov@gmail.com
Константин Олегович Коровин², korovin.ko@sut.ru

¹ООО «Специальный Технологический Центр»,
Санкт-Петербург, 195220, Российская Федерация

²Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича,
Санкт-Петербург, 193232, Российская Федерация

Аннотация: Конкретизирован интегро-дифференциальный оператор, определяющий продольную составляющую напряженности электрического поля на поверхности проводника излучателя несимметричного вибратора с вынесенной точкой питания (НВВТП), создаваемую базовым током (БТ). Получены аналитические выражения, определяющие параметры эквивалентного восьмиполюсника, являющегося внутренним сопротивлением эквивалентных генераторов БТ. Исследовано влияние длины проводников противовесов и верхней нагрузки НВВТП на коэффициент полезного действия и коэффициент усиления.

Ключевые слова: коэффициент трансформации, комплексная емкость, сопротивление излучения, входное сопротивление антенны, несимметричный вибратор, коэффициент усиления

Ссылка для цитирования: Попов О.В., Тумашов А.В., Борисов Г.Н., Коровин К.О. Математическая модель несимметричного вибратора с вынесенной точкой питания. Часть 3. Коэффициент усиления // Труды учебных заведений связи. 2023. Т. 9. № 6. С. 6–23. DOI:10.31854/1813-324X-2023-9-6-6-23

Mathematical Model of the Unbalanced Monopole Feed. Part 3. Gain

- Oleg Popov¹✉, ov.popov@mail.ru
Andrey Tumashov¹, ice47reg@yandex.ru
Georgy Borisov¹, georgiiborisov@gmail.com
Konstantin Korovin², korovin.ko@sut.ru

¹Special Technology Center LLC,
St. Petersburg, 195220, Russian Federation

²The Bonch-Bruевич Saint-Petersburg State University of Telecommunications,
St. Petersburg, 193232, Russian Federation

Abstract: An integro-differential operator is specified that determines the longitudinal component of the electric field strength on the surface of the emitter conductor of an Unbalanced Monopole with Shunt Feed, created by the Base Current (BC). Analytical expressions are obtained that determine the parameters of an equivalent eight pole, which is the internal resistance of equivalent BC generators. The influence of the length of the counterweight conductors and the upper load of the high-voltage transformer on the efficiency and gain was studied.

Keywords: transformation ratio, complex capacitance, radiation resistance, antenna input impedance, unbalanced monopole, gain

For citation: Popov O., Tumashov A., Borisov G., Korovin K. Mathematical Model of the Unbalanced Monopole Feed. Part 3. Gain. *Proceedings of Telecommun. Univ.* 2023;9(6):6–23. DOI:10.31854/1813-324X-2023-9-6-6-23

Введение

Сложность энергетического расчета радиотолков поверхностных волн заключается в трудности определения коэффициентов усиления антенн, устанавливаемых, как правило, непосредственно на полупроводящей поверхности [1]. Токи, наводимые в почве близкими полями, существенно снижают коэффициент полезного действия (КПД) антенны и, следовательно, ее коэффициент усиления (КУ). При этом, если дополнительное затухание сигнала, вызванное взаимодействием электромагнитных волн (ЭМВ) с подстилающей поверхностью при распространении в непосредственной близости от нее, хорошо описано в литературе [2–4] и представлено в виде формул и графиков, оценка КУ антенн, работающих поверхностной волной, производится приблизительно, без определения аналитической связи между электрическими параметрами почвы и КПД антенны [2, 5].

Как известно [2, 3, 6], под КПД антенны обычно понимают произведение коэффициента согласования по сопротивлению на КПД согласованной антенны. Таким образом, для определения КПД необходимо найти входное сопротивление антенны, представив при этом его вещественную составляющую в виде суммы сопротивления излучения и сопротивления тепловых потерь.

В первой части этой работы [7] построена развернутая эквивалентная схема несимметричного вибратора с вынесенной точкой питания (НВВТП), изображенная на рисунке 1. При построении схемы предполагалось, что все входящие в конструкцию НВВТП проводники можно разделить на две группы:

- проводники, непосредственно участвующие в процессе излучения (приема) ЭМВ;
- проводники неизлучающие (не принимающие) ЭМВ, но влияющие на распределение токов по излучающим (принимающим) ЭМВ и, таким образом, влияющие на характеристики антенны.

К проводникам первой группы относятся вертикальные соосные проводники, непосредственно примыкающие к узлу питания. Они названы излучателем НВВТП.

Эквивалентная схема представляет собой иллюстрацию формул, полученных с учетом допущений, принятых в методе наводимых электродвижущих сил (ЭДС):

- ток, протекающий по проводникам антенны, сосредоточен в осях проводников;
- распределение токов вдоль проводников описывается соотношениями теории линий.

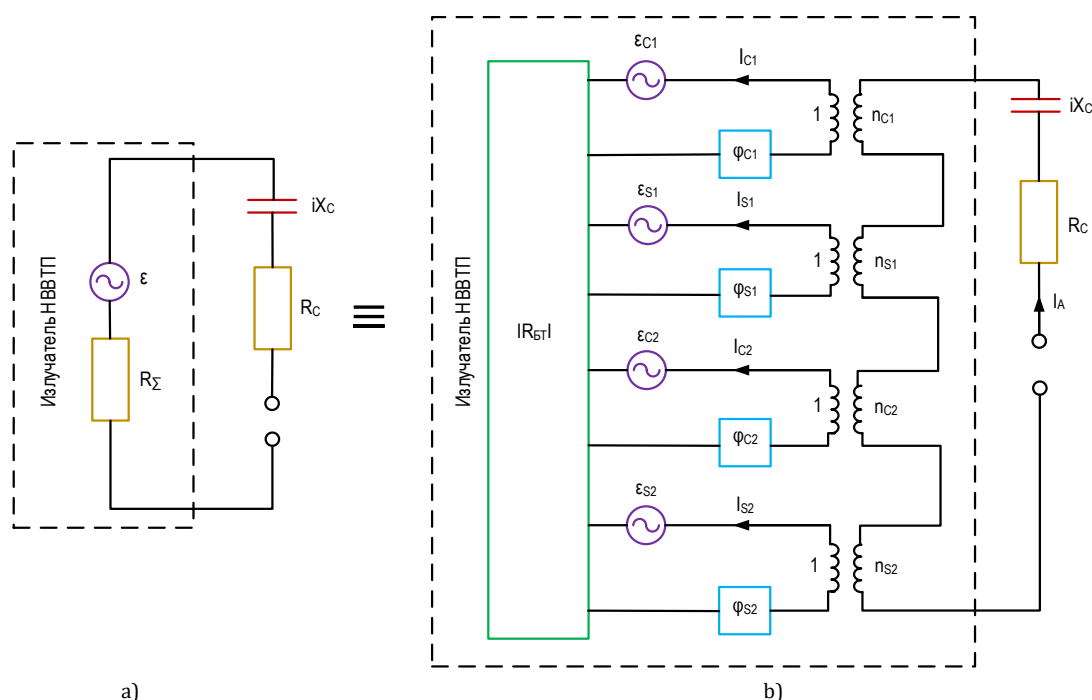


Рис. 1. Эквивалентные схемы НВВТП [7]

Fig. 1. Equivalent Circuits of UMSF [7]

Исходя из этих допущений, распределение тока по каждому из проводников излучателя описывается двумя вещественными гармоническими функциями, которые предложено называть базовыми, а ток, распределенный по закону базовой функции – базовым током (БТ). В связи с изложенным, распределение тока по проводникам излучателя в обобщенном виде будет [7]:

$$I_i(z_k) = I_a[n_{ci}j_{ci}^0(z_k) + n_{si}j_{si}^0(z_k)], \quad (1)$$

где

$$j_{c1}^0(z_k) = \begin{cases} 0 & 0 \leq z_k \leq l_{k2} \\ \cos(L_k - z_k) & l_{k2} \leq z_k \leq L_k' \end{cases} \quad (2)$$

$$j_{s1}^0(z_k) = \begin{cases} 0 & 0 \leq z_k \leq l_{k2} \\ \sin(L_k - z_k) & l_{k2} \leq z_k \leq L_k' \end{cases} \quad (3)$$

$$j_{c2}^0(z_k) = \begin{cases} \cos(z_k) & 0 \leq z_k \leq l_{k2} \\ 0 & l_{k2} \leq z_k \leq L_k' \end{cases} \quad (4)$$

$$j_{s2}^0(z_k) = \begin{cases} \sin(z_k) & 0 \leq z_k \leq l_{k2} \\ 0 & l_{k2} \leq z_k \leq L_k' \end{cases} \quad (5)$$

$I_i(z_k)$ – распределение тока по i -му проводнику излучателя, причем $i = 1$ соответствует верхнему проводнику, а $i = 2$ – нижнему; I_a – амплитуда тока на входных зажимах НВВТП; n_{ci} – коэффициент трансформации тока на зажимах антенны (КТТЗА) к пучности косинусного базового тока (ПКБТ) i -го проводника излучателя; n_{si} – КТТЗА к пучности синусного базового тока (ПСБТ) i -го проводника излучателя; $j_{c1}^0(z_k), j_{s1}^0(z_k), j_{c2}^0(z_k), j_{s2}^0(z_k)$ – базовые функции; $z_k = kz$ – электрическая координата точки на оси проводника излучателя; $k = 2\pi/\lambda$ – волновое число; λ – длина волны; z – координата точки на оси излучателя; $l_{ki} = kl_i$ – электрическая длина i -го проводника излучателя; l_i – длина i -го проводника излучателя; $L_k = l_{k1} + l_{k2}$ – электрическая длина излучателя.

Как показано в [7], КТТЗА к пучностям базовых токов (ПБТ) являются комплексными величинами. На развернутой эквивалентной схеме они изображены в виде соединений идеальных трансформаторов с фазовращателями. Аналитические выражения, определяющие КТТЗА к ПБТ, также приведены в [7]. Входящие в них значения комплексных емкостей НВВТП получены в [8], что позволяет считать КТТЗА к ПБТ известными.

Из рисунка 1 видно, что развернутая эквивалентная схема НВВТП содержит четыре генератора ЭДС БТ, подключенных ко входам восьмиполюсника внутренних сопротивлений. Взаимные сопротивления этого восьмиполюсника определяются соотношениями [7]:

$$\begin{aligned} r_{PP'} &= \operatorname{Re} \left[\frac{1}{k} \int_{l_{ki}} j_P^0(z_k) \bar{E}_i \{j_{P'}^0; z_k\} \bar{z}^0 dz_k \right] = \\ &= \frac{1}{k} \int_{l_{ki}} j_P^0(z_k) \operatorname{Re}(\bar{E}_i \{j_{P'}^0; z_k\} \bar{z}^0) dz_k, \end{aligned} \quad (6)$$

где $\bar{E}_i \{j_{P'}^0; z_k\} \bar{z}^0$ – линейный интегро-дифференциальный оператор, определяющий продольную напряженность электрического поля на поверхности i -го проводника в точке z_k , создаваемую единичным БТ P' ; $P = C1, S1, C2, S2$ – обобщенный индекс базовой функции; $\int_{l_{ki}}$ – означает интегрирование по длине i -го проводника излучателя при:

$$i = \begin{cases} 1 & \text{при } P = C1, S1 \\ 2 & \text{при } P = C2, S2 \end{cases}.$$

В (6) учтено, что функция $j_P^0(z_k)$ – вещественная.

Сопротивление излучения НВВТП, как видно из эквивалентной схемы и следует из преобразований, приведенных в [7], будет:

$$R_\Sigma = \sum_P \sum_{P'} n_P^* n_{P'} r_{PP'}. \quad (7)$$

Легко заметить, что сопротивление излучения не описывается аналитическими выражениями, поскольку соотношение (6) таковым не является. Для его приведения к аналитическому виду требуется, прежде всего, конкретизировать интегро-дифференциальный оператор $\bar{E}_i \{j_{P'}^0; z_k\} \bar{z}^0$.

Продольные составляющие напряженности электрического поля на поверхности излучателя, создаваемого базовыми токами

Поскольку излучатель НВВТП состоит из соосных вертикальных проводников, а электрическое поле на поверхности каждого из них создается совокупным воздействием четырех БТ, конкретизация оператора $\bar{E}_i \{j_{P'}^0; z_k\} \bar{z}^0$ состоит в определении вертикальных составляющих поля, создаваемых каждым из этих токов. Для достижения этой цели можно воспользоваться известным решением задачи о распределении вектора Герца поля элементарного вертикального диполя, размещенного на высоте ξ над полупроводящей поверхностью. Как известно [6, 9], вектор Герца такого диполя имеет только одну продольную составляющую, которая осесимметрично распределена в пространстве над полупроводящей поверхностью следующим образом:

$$d\Pi_z = \frac{30I_a dl_k}{ik} \Phi^-(r_k; \xi_k, z_k), \quad (8)$$

где

$$\Phi^-(r_k; \xi_k, z_k) = V(r_k; \xi_k - z_k) - W(r_k; \xi_k + z_k), \quad (9)$$

$$V(r_k, x_k) = \frac{e^{-i\sqrt{r_k^2 + x_k^2}}}{\sqrt{r_k^2 + x_k^2}}, \quad (10)$$

$$W(r_k, x_k) = V(r_k, x_k) - \Omega(r_k, x_k), \quad (11)$$

$$\Omega(r_k, x_k) = 2\varepsilon_k \int_0^\infty \frac{J_0(vr_k) e^{-x_k \sqrt{v^2 - 1}}}{\sqrt{v^2 - \varepsilon_k} + \varepsilon_k \sqrt{v^2 - 1}} v dv, \quad (12)$$

$I_a dl_k = I_a k dl$ – электрический момент диполя; $r_k = kr$ – электрическое расстояние от оси диполя до

точки наблюдения; $\xi_k = k\xi$ – электрическая высота диполя над полупроводящей поверхностью; $z_k = kz$ – электрическая высота точки наблюдения; z – координата точки наблюдения; $\varepsilon_k = \varepsilon - i60\sigma\lambda$ – комплексная относительная диэлектрическая проницаемость подстилающей поверхности; ε – относительная диэлектрическая проницаемость подстилающей поверхности; σ – удельная проводимость подстилающей поверхности.

Можно видеть, что функции $\Phi^-(r_k; \xi_k, z_k)$, $V(r_k; x_k)$, $W(r_k, x_k)$ и $\Omega(r_k, x_k)$ также, как и входящие в них параметры, являются безразмерными.

Для проведения дальнейших вычислений целесообразно определить значения функции $\Phi^-(r_k; \xi_k, z_k)$ при отсутствии границы раздела сред. В этом случае $\varepsilon_k = 1$, а соотношение (12) преобразуется к виду [9]:

$$\Omega(r_k, x_k) = \int_0^\infty \frac{J_0(vr_k)e^{-x_k\sqrt{v^2-1}}}{\sqrt{v^2-1}} v dv = V(r_k, x_k). \quad (13)$$

Тогда из соотношений (9) и (11) следует:

$$\left. \begin{aligned} W(r_k, x_k) &= 0 \\ \Phi^-(r_k; \xi_k, z_k) &= V(r_k; \xi_k - z_k) \end{aligned} \right\}. \quad (14)$$

Каждый БТ с единичной амплитудой в пучности можно представить бесконечным множеством элементарных вертикальных диполей, моменты которых зависят от высоты их размещения следующим образом:

$$I_d dl_k = j_{P'}^0(\xi_k) d\xi_k \text{ при } 0 \leq \xi_k \leq L_k. \quad (15)$$

Вертикальная составляющая напряженности электрического поля связана с вектором Герца вертикального электрического диполя, соотношением [6]:

$$dE_z = \left(k^2 + \frac{\partial^2}{\partial z^2}\right) d\Pi_z = k^2 \left(1 + \frac{\partial^2}{\partial z_k^2}\right) d\Pi_z. \quad (16)$$

Таким образом, электрическое поле, созданное единичным БТ, т. е. совокупным воздействием бесконечного числа диполей (15), будет:

$$E_z = \int_0^{L_k} \left(k^2 + \frac{\partial^2}{\partial z^2}\right) d\Pi_z = k^2 \int_0^{L_k} \left(1 + \frac{\partial^2}{\partial z_k^2}\right) d\Pi_z. \quad (17)$$

Выражение (17) с учетом (8) и (15) преобразуется к виду:

$$E_z = -30ik \int_0^{L_k} j_{P'}^0(\xi_k) \left(1 + \frac{\partial^2}{\partial z_k^2}\right) \times \Phi^-(r_k; \xi_k, z_k) d\xi_k. \quad (18)$$

С помощью соотношений (9–12) легко убедиться в справедливости равенства:

$$\frac{\partial^2}{\partial z_k^2} \Phi^-(r_k; \xi_k, z_k) = \frac{\partial^2}{\partial \xi_k^2} \Phi^-(r_k; \xi_k, z_k). \quad (19)$$

Выражение (19) преобразовывает (18) к виду:

$$E_z = -30ik \int_0^{L_k} j_{P'}^0(\xi_k) \left(1 + \frac{\partial^2}{\partial \xi_k^2}\right) \times \Phi^-(r_k; \xi_k, z_k) d\xi_k. \quad (20)$$

В соответствии с (2–5), вычисляя вторую производную по ξ от $j_{P'}^0(\xi)$, получим:

$$\frac{\partial^2}{\partial \xi_k^2} j_{P'}^0(\xi_k) = -j_{P'}^0(\xi_k). \quad (21)$$

Из правила интегрирования по частям, следует:

$$\begin{aligned} \int U d^2V &= U dV - \int dV dU = \\ &= U dV - V dU + \int V d^2U. \end{aligned} \quad (22)$$

Применение (21) и (22) к интегралу (20) позволяет вычислить его путем следующих преобразований (23).

$$\begin{aligned} E_z &= -30ik \left\{ \int_0^{L_k} j_{P'}^0(\xi_k) \Phi^-(r_k; \xi_k, z_k) d\xi_k + j_{P'}^0(\xi_k) \frac{\partial}{\partial \xi_k} \Phi^-(r_k; \xi_k, z_k) \Big|_0^{L_k} - \Phi^-(r_k; \xi_k, z_k) \frac{\partial}{\partial \xi_k} j_{P'}^0(\xi_k) \Big|_0^{L_k} - \right. \\ &\quad \left. - \int_0^{L_k} j_{P'}^0(\xi_k) \Phi^-(r_k; \xi_k, z_k) \frac{\partial}{\partial \xi_k} \right\} = -30ik \left[j_{P'}^0(\xi_k) \frac{\partial}{\partial \xi_k} \Phi^-(r_k; \xi_k, z_k) - \Phi^-(r_k; \xi_k, z_k) \frac{\partial}{\partial \xi_k} j_{P'}^0(\xi_k) \right] \Big|_0^{L_k}. \end{aligned} \quad (23)$$

Дифференцируя функции $V(r_k; \xi_k, z_k)$ и $W(r_k; \xi_k, z_k)$ по ξ_k и по z_k , с помощью соотношений (10–12), легко убедиться в справедливости следующих равенств:

$$\frac{\partial}{\partial \xi_k} V(r_k; \xi_k - z_k) = -\frac{\partial}{\partial z_k} V(r_k; \xi_k - z_k), \quad (24)$$

$$\frac{\partial}{\partial \xi_k} W(r_k; \xi_k + z_k) = \frac{\partial}{\partial z_k} W(r_k; \xi_k + z_k). \quad (25)$$

Из соотношений (9, 24 и 25) следует, что производная от функции $\Phi^-(r_k; \xi_k, z_k)$ по ξ_k может быть выражена через производные от функций $V(r_k; \xi_k - z_k)$ и $W(r_k; \xi_k + z_k)$ по z_k :

$$\begin{aligned} \frac{\partial}{\partial \xi_k} \Phi^-(r_k; \xi_k, z_k) &= -\frac{\partial}{\partial z_k} V(r_k; \xi_k - z_k) - \\ -\frac{\partial}{\partial z_k} W(r_k; \xi_k + z_k) &= -\frac{\partial}{\partial z_k} \Phi^+(r_k; \xi_k, z_k), \end{aligned} \quad (26)$$

где

$$\Phi^+(r_k; \xi_k, z_k) = V(r_k; \xi_k - z_k) + W(r_k; \xi_k + z_k). \quad (27)$$

Соотношения (23 и 26) позволяют легко получить выражение для вертикальной составляющей

напряженности электрического поля, создаваемого БТ P' , на поверхностях проводников.

Подставляя в (23) соответствующие базовые функции из соотношений (2–5) и учитывая области их задания, получим выражения (28–31), где $a_{ki} = ka_i$ – электрический радиус i -го проводника; a_i – радиус i -го проводника.

Подставляя в (28–31) пределы изменения переменной ξ_k , получим выражения (32–35).

$$\bar{E}_i\{j_{C1}^0; z_k\}\bar{z}^0 = 30ik \left[\cos(L_k - \xi_k) \frac{\partial}{\partial z_k} \Phi^+(a_{ki}; \xi_k, z_k) + \sin(L_k - \xi_k) \Phi^-(a_{ki}; \xi_k, z_k) \right] \Big|_{l_{k2}}^{L_k}, \quad (28)$$

$$\bar{E}_i\{j_{S1}^0; z_k\}\bar{z}^0 = 30ik \left[\sin(L_k - \xi_k) \frac{\partial}{\partial z_k} \Phi^+(a_{ki}; \xi_k, z_k) - \cos(L_k - \xi_k) \Phi^-(a_{ki}; \xi_k, z_k) \right] \Big|_{l_{k2}}^{L_k} \quad (29)$$

$$\bar{E}_i\{j_{C2}^0; z_k\}\bar{z}^0 = 30ik \left[\cos \xi_k \frac{\partial}{\partial z_k} \Phi^+(a_{ki}; \xi_k, z_k) - \sin \xi_k \Phi^-(a_{ki}; \xi_k, z_k) \right] \Big|_0^{l_{k2}}, \quad (30)$$

$$\bar{E}_i\{j_{S2}^0; z_k\}\bar{z}^0 = 30ik \left[\sin \xi_k \frac{\partial}{\partial z_k} \Phi^+(a_{ki}; \xi_k, z_k) + \cos \xi_k \Phi^-(a_{ki}; \xi_k, z_k) \right] \Big|_0^{l_{k2}}. \quad (31)$$

$$\bar{E}_i\{j_{C1}^0; z_k\}\bar{z}^0 = 30ik \left[\frac{\partial}{\partial z_k} \Phi^+(a_{ki}; L_k, z_k) - \cos l_{k1} \frac{\partial}{\partial z_k} \Phi^+(a_{ki}; l_{k2}, z_k) - \sin l_{k1} \Phi^-(a_{ki}; l_{k2}, z_k) \right], \quad (32)$$

$$\bar{E}_i\{j_{S1}^0; z_k\}\bar{z}^0 = 30ik \left[\cos l_{k1} \Phi^-(a_{ki}; l_{k2}, z_k) - \sin l_{k1} \frac{\partial}{\partial z_k} \Phi^+(a_{ki}; l_{k2}, z_k) - \Phi^-(a_{ki}; L_k, z_k) \right], \quad (33)$$

$$\bar{E}_i\{j_{C2}^0; z_k\}\bar{z}^0 = 30ik \left[\cos l_{k2} \frac{\partial}{\partial z_k} \Phi^+(a_{ki}; l_{k2}, z_k) - \frac{\partial}{\partial z_k} \Phi^+(a_{ki}; 0, z_k) - \sin l_{k2} \Phi^-(a_{ki}; l_{k2}, z_k) \right], \quad (34)$$

$$\bar{E}_i\{j_{S2}^0; z_k\}\bar{z}^0 = 30ik \left[\sin l_{k2} \frac{\partial}{\partial z_k} \Phi^+(a_{ki}; l_{k2}, z_k) + \cos l_{k2} \Phi^-(a_{ki}; l_{k2}, z_k) - \Phi^-(a_{ki}; 0, z_k) \right] \quad (35)$$

Можно видеть, что интегро-дифференциальный оператор $\bar{E}_i\{j_{P,i}^0; z_k\}\bar{z}^0$ состоит из трех слагаемых, каждое из которых содержит функцию $\Phi^+(a_{ki}; A_k, z_k)$, либо $\Phi^-(a_{ki}; A_k, z_k)$, либо их производную по z_k , где параметром A_k обозначена некоторая постоянная, принимающая значения 0, l_{k2} или L_k . В свою очередь упомянутые функции, согласно (9, 11, 27) состоят также из трех слагаемых, каждому из которых можно поставить в соответствие конкретный физический процесс.

При размещении антенны в однородной воздушной среде ($\epsilon_k = 1$), как следует из (14 и 27):

$$\begin{aligned} \Phi^+(a_{ki}; A_k, z_k) &= \Phi^-(a_{ki}; A_k, z_k) = \\ &= V(a_{ki}; A_k - z_k). \end{aligned} \quad (36)$$

Это соотношение позволяет рассматривать интегро-дифференциальный оператор $\bar{E}_i\{j_{P,i}^0; z_k\}\bar{z}^0$, как сумму двух интегро-дифференциальных операторов, определяющих напряженности падающей и отраженной волн, соответственно:

$$\bar{E}_i\{j_{P,i}^0; z_k\}\bar{z}^0 = \bar{E}_i^+\{j_{P,i}^0; z_k\}\bar{z}^0 + \bar{E}_i^-\{j_{P,i}^0; z_k\}\bar{z}^0, \quad (37)$$

где $\bar{E}_i^+\{j_{P,i}^0; z_k\}\bar{z}^0$ – интегро-дифференциальный оператор, определяющий вертикальную составляющую напряженности электрического поля, создаваемую БТ P' на поверхности i -го проводника, при отсутствии границы раздела сред (падающая волна); $\bar{E}_i^-\{j_{P,i}^0; z_k\}\bar{z}^0$ – интегро-дифференциальный оператор, определяющий вертикальную составляющую напряженности отраженного от границы раздела электрического поля, создаваемого БТ P' на поверхности i -го проводника (отраженная волна).

Учитывая, что в выражение (6) входит вещественная составляющая оператора $\bar{E}_i\{j_{P,i}^0; z_k\}\bar{z}^0$, представляют интерес вещественные составляющие операторов $\bar{E}_i^+\{j_{P,i}^0; z_k\}\bar{z}^0$ и $\bar{E}_i^-\{j_{P,i}^0; z_k\}\bar{z}^0$. Определить $\text{Re}(\bar{E}_i^+\{j_{P,i}^0; z_k\}\bar{z}^0)$ можно подставляя (36) в (32–35) и выделяя вещественные составляющие в полученных выражениях. Окончательным результатом являются соотношения (38–40), где $n = 1, 2$ – номер проводника с БТ, а $i = 1, 2$ – номер проводника, на поверхности которого определяется напряженность поля.

В (40) учтено, что $V(a_{ki}, x_k)$ задается соотношением (10), а также, что $a_{ki} \ll x_k$.

$$\operatorname{Re}[\bar{E}_i^+ \{j_{Cn}^0; z_k\} \bar{z}^0] = 30k \left\{ (-1)^n \frac{\partial}{\partial z_k} [\cos l_{kn} \operatorname{Riv}(a_{ki}, l_{k2} - z_k) - \operatorname{Riv}(a_{ki}, \delta_n^1 L_k - z_k)] - \sin l_{kn} \operatorname{Riv}(a_{ki}, l_{k2} - z_k) \right\}. \quad (38)$$

$$\operatorname{Re}[\bar{E}_i^+ \{j_{Sn}^0; z_k\} \bar{z}^0] = 30k \left\{ \cos l_{kn} \operatorname{Riv}(a_{ki}, l_{k2} - z_k) - \operatorname{Riv}(a_{ki}, \delta_n^1 L_k - z_k) + (-1)^n \sin l_{kn} \frac{\partial}{\partial z_k} \operatorname{Riv}(a_{ki}, l_{k2} - z_k) \right\}, \quad (39)$$

где

$$\operatorname{Riv}(a_{ki}, x_k) = \operatorname{Re}[iV(a_{ki}, x_k)] = \frac{\sin \sqrt{a_{ki}^2 + x_k^2}}{\sqrt{a_{ki}^2 + x_k^2}} \approx \frac{\sin x_k}{x_k}, \quad (40)$$

$$x_k = -z_k, l_{k2} - z_k, L_k - z_k, \delta_n^1 = \begin{cases} 1 & \text{при } n = 1 \\ 0 & \text{при } n \neq 1 \end{cases} - \text{символ Кронекера [10].}$$

Подставляя в (32–35) соотношения (9 и 27) без первого слагаемого и выделяя вещественные составляющие, получим выражения (41–44), определяющие оператор $\operatorname{Re}(\bar{E}_i^+ \{j_{Pn}^0; z_k\} \bar{z}^0)$. В (41, 42), также, как и в (38, 39) $n = 1, 2$ – номер проводника с БТ, а $i = 1, 2$ – номер проводника, на поверхности которого определяется напряженность поля отраженного от полупроводящей земли.

Можно видеть, что вещественные составляющие напряженности поля отраженной волны определяются мнимой составляющей функции $\Omega(a_{ki}, x_k)$ и ее производными по x_k . Сама функция $\Omega(a_{ki}, x_k)$ представляет собой несобственный интеграл (12), сходимость которого зависит от координат точки

наблюдения и параметров почвы. Т.е. возникает задача выделения из $\Omega(a_{ki}, x_k)$ мнимой составляющей, исследования скорости ее сходимости и получения аналитических выражений, пригодных для инженерных расчетов.

Преобразование функции $\operatorname{Im}[\Omega(a_{ki}, x_k)]$ к виду, пригодному для инженерных расчетов

Для разделения несобственного интеграла (12) на вещественную и мнимую составляющие, представим его в виде суммы двух интегралов, предварительно сделав в нем замену переменной $t = \nu a_{ki}$. Получим выражение (44).

$$\operatorname{Re}[\bar{E}_i^- \{j_{Cn}^0; z_k\} \bar{z}^0] = 30k \left\{ (-1)^n \frac{\partial}{\partial z_k} [\cos l_{kn} \operatorname{Riw}(a_{ki}, l_{k2} + z_k) - \operatorname{Riw}(a_{ki}, \delta_n^1 L_k + z_k)] + \sin l_{kn} \operatorname{Riw}(a_{ki}, l_{k2} + z_k) \right\}, \quad (41)$$

$$\operatorname{Re}[\bar{E}_i^- \{j_{Sn}^0; z_k\} \bar{z}^0] = 30k \left\{ (-1)^n \sin l_{kn} \frac{\partial}{\partial z_k} \operatorname{Riw}(a_{ki}, l_{k2} + z_k) - \cos l_{kn} \operatorname{Riw}(a_{ki}, l_{k2} + z_k) + \operatorname{Riw}(a_{ki}, \delta_n^1 L_k + z_k) \right\}, \quad (42)$$

где

$$\operatorname{Riw}(a_{ki}, x_k) = \operatorname{Re}[iW(a_{ki}, x_k)] = \operatorname{Riv}(x_k) + \operatorname{Im}[\Omega(a_{ki}, x_k)], \quad (43)$$

$$x_k = z_k, l_{k2} + z_k, L_k + z_k.$$

$$\Omega(a_{ki}, x_k) = \frac{2\varepsilon_k}{a_{ki}} \int_0^\infty \frac{J_0(t) e^{-x_k t_{ai}}}{\sqrt{t^2 - a_{ki}^2 \varepsilon_k} + \varepsilon_k t_{ai} a_{ki}} t dt = \frac{2}{a_{ki}} \left[\int_0^{a_{ki}} \frac{e^{-i x_k t_{mai}}}{\frac{\sqrt{t^2 - a_{ki}^2 \varepsilon_k}}{\varepsilon_k} + i t_{mai} a_{ki}} t dt + \int_{a_{ki}}^\infty \frac{J_0(t) e^{-x_k t_{ai}}}{\frac{\sqrt{t^2 - a_{ki}^2 \varepsilon_k}}{\varepsilon_k} + t_{ai} a_{ki}} t dt \right], \quad (44)$$

где

$$t_{ai} = \sqrt{t^2 - a_{ki}^2} / a_{ki}, t_{mai} = |t_{ai}| = \sqrt{a_{ki}^2 - t^2} / a_{ki} \text{ при } t \leq a_{ki}$$

В (44) учтено, что при $t \leq a_{ki}$ $J_0(t) \approx 1$, поскольку $a_{ki} \ll 1$. Легко показать, что:

$$(\sqrt{t^2 - a_{ki}^2 \varepsilon_k}) / \varepsilon_k = T e^{i(\Phi_1 - \Phi_2)}, \quad (45)$$

где

$$T = \frac{\sqrt[4]{(t^2 - a_{ki}^2 \varepsilon_k)^2 + (60\sigma\lambda)^2 a_{ki}^4}}{\sqrt{\varepsilon_k^2 + (60\sigma\lambda)^2}} = \frac{1}{|\varepsilon_k|} \sqrt[4]{t^4 - 2t^2 a_{ki}^2 \varepsilon_k + a_{ki}^4 |\varepsilon_k|^2}, \quad (46)$$

$$|\varepsilon_k|^2 = \varepsilon^2 + (60\sigma\lambda)^2, \quad (47)$$

$$\phi_1 = 0,5 \arctg(60\sigma\lambda a_{ki}^2 / (t^2 - a_{ki}^2 \varepsilon)), \quad (48)$$

$$\phi_2 = \arctg(-60\sigma\lambda/\varepsilon). \quad (49)$$

В (46, 47) $|\varepsilon_k|$ – модуль комплексной диэлектрической проницаемости земли.

Подставляя (45–49) в (44), можно выделить (50) вещественную и мнимую составляющие $\Omega(a_i, x_k)$.

$$\Omega(a_i, x_k) = \frac{2}{a_{ki}} \left\{ \int_0^{a_{ki}} \frac{\cos(x_k t_{mai} + \psi_1)}{F_1} t dt + \int_{a_{ki}}^{\infty} \frac{J_0(t) e^{-x_k t_{ai}} \cos \psi_2}{F_2} t dt - \right. \\ \left. - i \left[\int_0^{a_{ki}} \frac{\sin(x_k t_{mai} + \psi_1)}{F_1} t dt + \int_{a_{ki}}^{\infty} \frac{J_0(t) e^{-x_k t_{ai}} \sin \psi_2}{F_2} t dt \right] \right\}. \quad (50)$$

В выражении (50):

$$F_1 = \sqrt{T^2 + 2T\sqrt{a_{ki}^2 - t^2} \sin \Delta\phi + a_{ki}^2 - t^2}, \quad (51)$$

$$F_2 = \sqrt{T^2 + 2T\sqrt{t^2 - a_{ki}^2} \cos \Delta\phi - a_{ki}^2 + t^2}, \quad (52)$$

$$\psi_1 = \arctg[(T \sin \Delta\phi + t_{mai} a_{ki}) / (T \cos \Delta\phi)], \quad (53)$$

$$\psi_2 = \arctg[(T \sin \Delta\phi) / (T \cos \Delta\phi + t_{ai} a_{ki})], \quad (54)$$

$$\Delta\phi = \phi_1 - \phi_2. \quad (55)$$

Из соотношения (50) следует соотношение (56).

$$\operatorname{Im}[\Omega(a_{ki}, x_k)] = -\frac{2}{a_{ki}} \left[\int_0^{a_{ki}} \Phi_1(a_{ki}, x_k; t) dt + \int_{a_{ki}}^{\infty} \Phi_2(a_{ki}, x_k; t) dt \right], \quad (56)$$

где

$$\Phi_1(a_{ki}, x_k; t) = \frac{t}{F_1} \sin(x_k t_{mai} + \psi_1) = \frac{t}{F_1} [\sin(x_k t_{mai}) \cos \psi_1 + \sin \psi_1 \cos(x_k t_{mai})], \quad (57)$$

$$\Phi_2(a_{ki}, x_k; t) = J_0(t) \frac{t}{F_2} \exp(-x_k t_{ai}) \sin \psi_2. \quad (58)$$

Легко показать, что из (53 и 54) следует:

$$\left. \begin{aligned} \sin \psi_1 &= (T \sin \Delta\phi + \sqrt{a_{ki}^2 - t^2}) / F_1 \\ \cos \psi_1 &= T \cos \Delta\phi / F_1 \\ \sin \psi_2 &= T \sin \Delta\phi / F_2 \end{aligned} \right\}. \quad (59)$$

Соотношения (59) позволяют преобразовать (57 и 58) к следующему виду:

$$\Phi_1(a_{ki}, x_k; t) = \frac{t}{F_1^2} \times \quad (60)$$

$$\times [T \sin(x_k t_{mai} + \Delta\phi) + t_{mai} a_{ki} \cos(x_k t_{mai})],$$

$$\Phi_2(a_{ki}, x_k; t) = J_0(t) \frac{tT}{F_2^2} \exp(-x_k t_{ai}) \sin \Delta\phi. \quad (61)$$

Для вычисления $\operatorname{Im}[\Omega(a_{ki}, x_k)]$, несобственный интеграл в выражении (56) удобно представить в виде суммы бесконечного знакопеременного ряда, состоящего из определенных интегралов, пределы интегрирования которых совпадают с нулями функции Бесселя. Поскольку $\Phi_2(a_{ki}, x_k; t)$ при неограниченном возрастании t стремится к нулю, этот ряд будет сходящимся. Как известно [11], погрешность вычисления суммы такого ряда не превышает первого отброшенного члена, что позволяет легко определить, при каком x/a_i будет обеспечиваться приемлемая для инженерных расчетов

точность вычисления, если верхний предел несобственного интеграла ограничить первым нулем функции $J_0(t)$.

Пусть:

$$\operatorname{Im}[\Omega(a_{ki}, x_k)] \approx -\frac{2}{a_{ki}} \times \left[\int_0^{a_{ki}} \Phi_1(a_{ki}, x_k; t) dt + \int_{a_{ki}}^{j_{01}} \Phi_2(a_{ki}, x_k; t) dt \right], \quad (62)$$

где $j_{01} = 2,4048$ – первый ноль функции Бесселя нулевого порядка [12].

Относительная погрешность вычисления в этом случае будет:

$$\zeta = \frac{\left| \int_{j_{01}}^{j_{02}} \Phi_2(a_{ki}, x_k; t) dt \right|}{\left| \int_0^{a_{ki}} \Phi_1(a_{ki}, x_k; t) dt + \int_{a_{ki}}^{j_{01}} \Phi_2(a_{ki}, x_k; t) dt \right|},$$

где $j_{02} = 5,5201$ – второй ноль функции Бесселя нулевого порядка.

На рисунке 2а представлено семейство кривых, характеризующих зависимость относительной погрешности вычислений от x/a_i , для различных частот и удельных проводимостей почвы. Видно, что при $x/a_i > 2$ относительная погрешность не превышает 0,01.

На рисунке 2b – те же зависимости для случая, когда верхний предел несобственного интеграла уменьшен до одной десятой j_{01} . Видно, что даже такое существенное ограничение верхнего предела является целесообразным при $x/a_i > 20$, поскольку позволяет, сохраняя точность вычислений, значительно сократить их объем. Это объясняется не только сокращением интервала интегрирования, а и возможностью упростить подынтегральную функцию. Действительно, при $0 \leq t \leq 0,24$ значения функции Бесселя нулевого порядка лежат в пределах $1 \geq J_0(t) \geq 0,986$, что позволяет исключить ее из подынтегрального выражения. Пусть:

$$\Phi_2^{(0)}(a_{ki}, x_k; t) = \frac{tT}{F_2^2} \exp(-x_k t_{ai}) \sin \Delta \Phi. \quad (63)$$

Тогда, при $t \leq 0,24$ справедливо приближение:

$$\int_{a_{ki}}^{0,24} \Phi_2(a_{ki}, x_k; t) dt \approx \int_{a_{ki}}^{0,24} \Phi_2^{(0)}(a_{ki}, x_k; t) dt. \quad (63a)$$

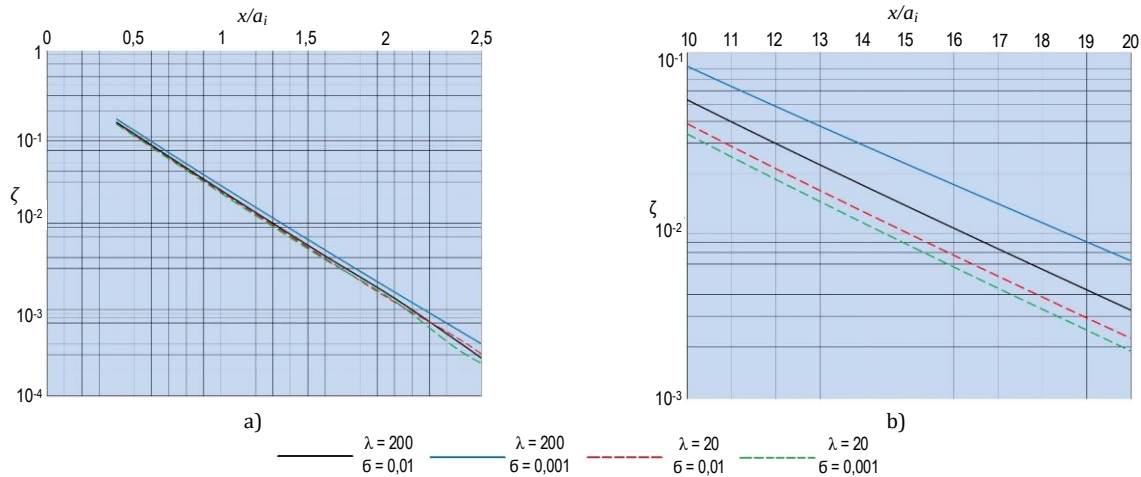


Рис. 2. Зависимость относительной погрешности вычисления $\text{Im}[\Omega(a_{ki}, x_k)]$ от x/a_i при верхнем пределе интегрирования: а) 2,4; б) 0,24

Fig. 2. Dependence of the Relative Calculation Error on $\text{Im}[\Omega(a_{ki}, x_k)]$ at the Upper Limit of Integration: a) 2.4; b) 0.24

$$Riw(a_{ki}, x_k) = \frac{\sin x_k}{x_k} - \frac{2}{a_{ki}} \left[\int_0^{a_{ki}} \Phi_1(a_{ki}, x_k; t) dt + F(a_{ki}, x_k) \right], \quad (64)$$

где

$$F(a_{ki}, x_k) = \int_{a_{ki}}^{\infty} \Phi_2(a_{ki}, x_k; t) dt = \begin{cases} \int_{0,24}^{2,4} \Phi_2(a_{ki}, x_k; t) dt & \text{при } x/a_i > 2 \\ \int_{a_{ki}}^{0,24} \Phi_2^{(0)}(a_{ki}, x_k; t) dt & \text{при } x/a_i > 20 \end{cases}, \quad (64a)$$

$$\text{Re}[i\Omega(a_{ki}, 0)] \approx \frac{2}{a_{ki}} \left[\int_0^{a_{ki}} \Phi_1(a_{ki}, 0; t) dt + \int_{a_{ki}}^{t_m} \Phi_2(a_{ki}, 0; t) dt + \int_{t_m}^{\infty} \Phi_2^{(a)}(a_{ki}, 0; t) dt \right]. \quad (65)$$

$$J_0(t) \approx \sqrt{\frac{2}{\pi t}} \cos\left(t - \frac{\pi}{4}\right). \quad (66)$$

Относительная погрешность вычислений, представленная на рисунке 2b, определялась следующим образом:

$$\zeta' = \frac{\left| \int_{0,24}^{2,4} \Phi_2(a_{ki}, x_k; t) dt \right|}{\left| \int_0^{a_{ki}} \Phi_1(a_{ki}, x_k; t) dt + \int_{a_{ki}}^{0,24} \Phi_2^{(0)}(a_{ki}, x_k; t) dt \right|}.$$

Таким образом, соотношение (43) с учетом (40, 56, 62 и 63) можно представить в виде (64). Как видно из (61), с уменьшением x_k скорость сходимости несобственного интеграла в (56) уменьшается, что и нашло отражение в (64a). Очевидно, что самой низкой будет сходимость при $x_k = 0$. В этом случае (56) удобно представить в виде суммы трех интегралов (65), где $\Phi_2^{(a)}(a_{ki}, 0; t)$ – асимптотическое приближение функции $\Phi_2(a_{ki}, 0; t)$ при $t \gg 1$.

Как известно [12], функцию Бесселя нулевого порядка при больших значениях аргумента можно представить в виде (66).

Из соотношений (46, 48 и 52) при $t \gg t_m$ и $a_{ki} \ll \ll 1$, следует:

$$T \approx \frac{t}{|\varepsilon_k|}, \Phi_1 \approx 0, \sin \Delta \Phi \approx \frac{60\sigma\lambda}{|\varepsilon_k|}, \cos \Delta \Phi \approx \frac{\varepsilon}{|\varepsilon_k|}, \quad (67)$$

$$F_2 \approx \frac{t}{|\varepsilon_k|} \sqrt{1 + 2\varepsilon + |\varepsilon_k|^2}.$$

Подставив (66 и 67) в (61), получим асимптотическое приближение функции $\Phi_2(a_{ki}, 0; t)$ при $t \gg 1$:

$$\Phi_2^{(a)}(a_{ki}, 0; t) = \sqrt{\frac{2}{\pi}} \frac{60\sigma\lambda}{1 + 2\varepsilon + |\varepsilon_k|^2} \frac{\cos\left(t - \frac{\pi}{4}\right)}{\sqrt{t}} = \quad (68)$$

$$= \frac{Q}{\sqrt{\pi}} \left[\frac{\cos t}{\sqrt{t}} + \frac{\sin t}{\sqrt{t}} \right],$$

где $Q = 60\sigma\lambda/(1 + 2\varepsilon + |\varepsilon_k|^2)$.

Соотношение (68) преобразует последнее слагаемое в правой части (65) к виду:

$$\int_{t_m}^{\infty} \Phi_2(a_{ki}, 0; t) dt = \int_{t_m}^{\infty} \Phi_2^{(a)}(a_{ki}, 0; t) dt = \quad (69)$$

$$\frac{Q}{\sqrt{\pi}} \left[\int_{t_m}^{\infty} \frac{\cos(t)}{\sqrt{t}} dt + \int_{t_m}^{\infty} \frac{\sin(t)}{\sqrt{t}} dt \right].$$

Учитывая, что [13]:

$$\int_0^{\infty} \frac{\sin(t)}{\sqrt{t}} dt = \int_0^{\infty} \frac{\cos(t)}{\sqrt{t}} dt = \sqrt{\frac{\pi}{2}}, \quad (70)$$

соотношение (69) легко свести к совокупности интегралов Френеля [12]:

$$\int_{t_m}^{\infty} \Phi_2^{(a)}(a_{ki}, 0; t) dt = \sqrt{2}Q[1 - C(t_m) - S(t_m)], \quad (71)$$

где $C(t_m)$ и $S(t_m)$ – интегралы Френеля:

$$C(t_m) = \frac{1}{\sqrt{2\pi}} \int_0^{t_m} \frac{\cos(t)}{\sqrt{t}} dt, \quad (72)$$

$$S(t_m) = \frac{1}{\sqrt{2\pi}} \int_0^{t_m} \frac{\sin(t)}{\sqrt{t}} dt.$$

$$Riw(a_{ki}, 0) \approx 1 - \frac{2}{a_{ki}} \left[\int_0^{a_{ki}} \Phi_1(a_{ki}, 0; t) dt + \int_{a_{ki}}^6 \Phi_2(a_{ki}, 0; t) dt + 0,29Q \right]. \quad (74)$$

Эти формулы будут весьма полезны при определении взаимных сопротивлений нагрузочного восьмиполюсника эквивалентных генераторов БТ (рисунок 1б). Подставляя (37) в (6), получим:

$$r_{PP'} = r_{PP'}^+ + r_{PP'}^-, \quad (75)$$

где

$$r_{PP'}^+ = \frac{1}{k} \int_{l_{ki}} j_P^0(z_k) \operatorname{Re}[\bar{E}_i^+ \{j_{P'}^0; z_k\} \bar{z}^0] dz_k, \quad (76)$$

Выбор рационального значения параметра t_m в (65) можно сделать на основе анализа среднеквадратического отклонения на интервале 2π асимптотического приближения подынтегральной функции (68) от своего точного значения при различных значениях t_m [14].

Обозначим:

$$S(t_m) = \sqrt{\frac{1}{2\pi} \int_{t_m}^{t_m+2\pi} \left[\frac{\Phi_2(a_{ki}, 0; t) - \Phi_2^{(a)}(a_{ki}, 0; t)}{M} \right]^2 dt}, \quad (73)$$

где M – максимальное значение $|\Phi_2(a_{ki}, 0; t)|$ на интервале интегрирования.

Графически зависимость $S(t_m)$ представлена на рисунке 3. Видно, что при $t_m > 6$ среднеквадратическое отклонение не превышает 0,01, что позволяет считать совпадение исходной функции с приближенной (69) достаточным для инженерных расчетов.

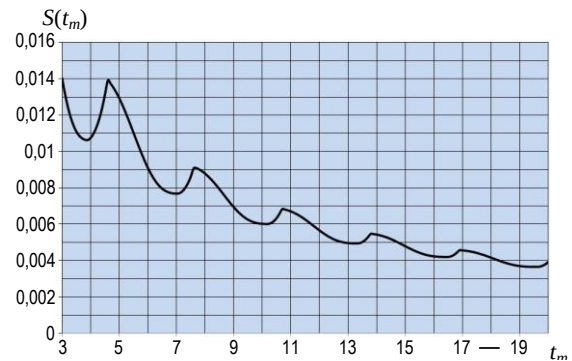


Рис. 3. Среднеквадратическое отклонение $\Phi_2^{(a)}(a_{ki}, 0; t)/M$ от $\Phi_2(a_{ki}, 0; t)/M$

Fig. 3. Standard Deviation $\Phi_2^{(a)}(a_{ki}, 0; t)/M$ from $\Phi_2(a_{ki}, 0; t)/M$

Соотношения (65 и 71) позволяют представить (43) при $x_{ai} = 0$ в виде (74), где учтено, что $C(6) = 0,4433$, $S(6) = 0,3499$ [12]. Таким образом, функцию $Riw(a_{ki}, x_k)$, содержащую несобственный интеграл с бесконечным верхним пределом, удалось представить в виде соотношений (64 и 74), содержащих простые определенные интегралы, легко поддающиеся численному интегрированию.

$$r_{PP'}^- = \frac{1}{k} \int_{l_{ki}} j_P^0(z_k) \operatorname{Re}[\bar{E}_i^- \{j_{P'}^0; z_k\} \bar{z}^0] dz_k, \quad (77)$$

$r_{PP'}^+$ – взаимное сопротивление восьмиполюсника внутренних сопротивлений БТ при отсутствии подстилающей поверхности; $r_{PP'}^-$ – поправка к взаимному сопротивлению восьмиполюсника внутренних сопротивлений БТ, вносимая границей раздела сред.

Легко заметить, что при отсутствии подстилающей поверхности НВВТП представляет собой несимметричный диполь с оконечными нагрузками (НДОН), сопротивление излучения которого будет определяться соотношением (7) с учетом того, что в однородной среде $r_{PP_i} = r_{PP_i}^+$. Таким образом, решение задачи о сопротивлении излучения НДОН можно рассматривать как первый этап определения сопротивления излучения НВВТП.

Сопротивление излучения

Соотношение (38) определяет вещественную составляющую вертикальной компоненты напряженности электрического поля, создаваемого косинусным БТ, протекающим по n -му проводнику, в точке с координатой z_k , лежащей на поверхности i -го проводника излучателя. Подставляя (38) в (76), получим взаимное сопротивление P -го БТ i -го проводника и косинусного БТ n -го проводника. Поскольку $Riv(a_{ki}, z_k) = Riv(z_k)$, как следует из (40), получим (78).

Подставляя в (76) соотношение (39), можно определить взаимное сопротивление P -го БТ i -го проводника и синусного БТ, протекающего по n -му проводнику (79).

Применив к первым интегралам соотношений (78 и 79) правило интегрирования по частям и опустив несложные, но громоздкие преобразования, с учетом обозначений (2–5) и соотношения (40), получим (80–89).

Для определения поправок к вещественным составляющим взаимных сопротивлений, вносимых границей раздела сред, следует воспользоваться соотношениями (41, 42 и 77). Подставив в (77) интегро-дифференциальный оператор (41), получим поправку к вещественной составляющей взаимного сопротивления между P -м БТ i -го проводника и косинусным БТ n -го проводника. Считая, что проводники излучателя НВВТП имеют одинаковое сечение, получим (91).

Подставив в (77) дифференциальный оператор (42), определим поправку к вещественной составляющей взаимного сопротивления P -го БТ i -го проводника и синусного БТ, протекающего по n -му проводнику (92).

Применив к первым интегралам соотношений (91 и 92) правило интегрирования по частям и опустив несложные, но громоздкие преобразования, с учетом обозначений (2–5) и соотношений (43 и 56), получим (93–102).

$$r_{PCn}^+ = 30 \left\{ (-1)^n \int_{l_{ki}} j_P^0(z_k) \frac{\partial}{\partial z_k} [Riv(\delta_n^1 L_k - z_k) - \cos l_{kn} Riv(l_{k2} - z_k)] dz_k + \right. \\ \left. + \sin l_{kn} \int_{l_{ki}} j_P^0(z_k) Riv(l_{k2} - z_k) dz_k \right\}. \quad (78)$$

$$r_{PSn}^+ = 30 \left\{ (-1)^n \sin l_{kn} \int_{l_{ki}} j_P^0(z_k) \frac{\partial}{\partial z_k} Riv(l_{k2} - z_k) dz_k + \right. \\ \left. + \cos l_{kn} \int_{l_{ki}} j_P^0(z_k) Riv(l_{k2} - z_k) dz_k - \int_{l_{ki}} j_P^0(z_k) Riv(\delta_n^1 L_k - z_k) dz_k \right\}. \quad (79)$$

$$r_{C1C1}^+ = 20 l_{k1}^2 \left[1 - \frac{3}{8} l_{k1}^2 \right], \quad (80)$$

$$r_{C1S1}^+ = r_{S1C1}^+ = 10 l_{k1}^3 \left[1 - \frac{4}{15} l_{k1}^2 \right], \quad (81)$$

$$r_{C1C2}^+ = r_{C2C1}^+ = 30 \left[IN1 - \frac{l_{k1} l_{k2}}{3} + \frac{l_{k1}^2 l_{k2}^2}{6} + \frac{l_{k1}^4 + l_{k2}^4}{30} \right], \quad (82)$$

$$r_{C1S2}^+ = r_{S2C1}^+ = 30 \left[IN2 - \frac{l_{k1}^2 l_{k2}}{3} \left(1 - \frac{l_{k1}^2}{10} - \frac{l_{k2}^2}{6} \right) \right], \quad (83)$$

$$r_{S1S1}^+ = 5 l_{k1}^4 \left[1 - \frac{1}{6} l_{k1}^2 \right], \quad (84)$$

$$r_{S1C2}^+ = r_{C2S1}^+ = 30 \left[IN2 - \frac{l_{k1} l_{k2}^2}{3} \left(1 - \frac{l_{k1}^2}{6} - \frac{l_{k2}^2}{10} \right) \right], \quad (85)$$

$$r_{S1S2}^+ = r_{S2S1}^+ = 30 \left[l_{k1} l_{k2} \left(1 - \frac{l_{k1}^2 + l_{k2}^2}{6} \right) - IN1 \right], \quad (86)$$

$$r_{C2C2}^+ = 20l_{k2}^2 \left[1 - \frac{3}{8} l_{k2}^2 \right], \quad (87)$$

$$r_{C2S2}^+ = r_{S2C2}^+ = 10l_{k2}^3 \left[1 - \frac{4l_{k2}^2}{15} \right], \quad (88)$$

$$r_{S2S2}^+ = 5 l_{k2}^4 \left[1 - \frac{l_{k2}^2}{6} \right], \quad (89)$$

где

$$\left. \begin{aligned} IN1 &= \cos L_k IN3 - \sin L_k IN4 \\ IN2 &= \cos L_k IN4 + \sin L_k IN3 \\ IN3 &= \frac{1}{2}(L_k^2 - l_{k1}^2 - l_{k2}^2) - \frac{1}{12}(L_k^4 - l_{k1}^4 - l_{k2}^4) + \frac{1}{240}(L_k^6 - l_{k1}^6 - l_{k2}^6) \\ IN4 &= -\frac{2}{9}(L_k^3 - l_{k1}^3 - l_{k2}^3) + \frac{1}{36}(L_k^5 - l_{k1}^5 - l_{k2}^5) \end{aligned} \right\}. \quad (90)$$

$$r_{PCn}^- = 30 \left\{ (-1)^n \int_{l_{ki}} j_P^0(z_k) \frac{\partial}{\partial z_k} [\cos l_{kn} \text{Riw}(a_k, l_{k2} + z_k) - \text{Riw}(a_k, \delta_n^1 L_k + z_k)] dz_k + \right. \\ \left. + \sin l_{kn} \int_{l_{ki}} j_P^0(z_k) \text{Riw}(a_k, l_{k2} - z_k) dz_k \right\}. \quad (91)$$

$$r_{PSn}^- = 30 \left\{ (-1)^n \sin l_{kn} \int_{l_{ki}} j_P^0(z_k) \frac{\partial}{\partial z_k} \text{Riw}(a_k, l_{k2} + z_k) dz_k - \right. \\ \left. - \cos l_{kn} \int_{l_{ki}} j_P^0(z_k) \text{Riw}(a_k, l_{k2} + z_k) dz_k + \int_{l_{ki}} j_P^0(z_k) \text{Riw}(a_k, \delta_n^1 L_k + z_k) dz_k \right\}. \quad (92)$$

$$r_{C1C1}^- = -30i [\text{Riw}(a_k, 2L_k) - \cos l_{k1} \text{Riw}(a_k, L_k + l_{k2}) - \cos l_{k1} A3 + \cos 2L_k IN5 - \sin 2L_k IN6], \quad (93)$$

$$r_{C1S1}^- = r_{S1C1}^- = -30i [-\sin l_{k1} A3 + \sin 2L_k IN5 + \cos 2L_k IN6], \quad (94)$$

$$r_{C1C2}^- = r_{C2C1}^- = -30i [\cos l_{k2} A3 - A4 + \cos L_k IN7 - \sin L_k IN8], \quad (95)$$

$$r_{C1S2}^- = r_{S2C1}^- = -30i [\sin l_{k2} A3 - \sin L_k IN7 - \cos L_k IN8], \quad (96)$$

$$r_{S1S1}^- = -30i [\sin^2 l_{k1} \text{Riw}(a_k, 2l_{k2}) - \cos 2L_k IN5 + \sin 2L_k IN6], \quad (97)$$

$$r_{S1C2}^- = r_{C2S1}^- = -30i [\sin l_{k1} A5 + \sin L_k IN7 + \cos L_k IN8], \quad (98)$$

$$r_{S1S2}^- = r_{S2S1}^- = -30i [-\sin l_{k1} \sin l_{k2} \text{Riw}(a_k, 2l_{k2}) + \cos L_k IN7 - \sin L_k IN8] \quad (99)$$

$$r_{C2C2}^- = -30i \{ \text{Riw}(a_k, 0) - \cos l_{k2} [A5 + \text{Riw}(a_k, l_{k2})] + IN9 \}, \quad (100)$$

$$r_{C2S2}^- = r_{S2C2}^- = -30i [\sin l_{k2} A5 + IN10], \quad (101)$$

$$r_{S2S2}^- = -30i [\sin^2 l_{k2} \text{Riw}(a_k, 2l_{k2}) - IN9]. \quad (102)$$

В соотношениях (93–102) использованы следующие обозначения:

$$\left. \begin{aligned} A3 &= \text{Riw}(a_k, L_k + l_{k2}) - \cos l_{k1} \text{Riw}(a_k, 2l_{k2}) \\ A4 &= \text{Riw}(a_k, L_k) - \cos l_{k1} \text{Riw}(a_k, l_{k2}) \\ A5 &= \text{Riw}(a_k, l_{k2}) - \cos l_{k2} \text{Riw}(a_k, 2l_{k2}) \end{aligned} \right\}, \quad (103)$$

$$\left. \begin{aligned} IN5 &= Iws(L_k + l_{k2}, 2L_k) - Iws(2l_{k2}, L_k + l_{k2}) \\ IN6 &= Iwc(L_k + l_{k2}, 2L_k) - Iwc(2l_{k2}, L_k + l_{k2}) \\ IN7 &= Iws(2l_{k2}, L_k + l_{k2}) - Iws(l_{k2}, L_k) \\ IN8 &= Iwc(2l_{k2}, L_k + l_{k2}) - Iwc(l_{k2}, L_k) \\ IN9 &= Iws(l_{k2}, 2l_{k2}) - Iws(0, l_{k2}) \\ IN10 &= Iwc(l_{k2}, 2l_{k2}) - Iwc(0, l_{k2}). \end{aligned} \right\}, \quad (104)$$

$$\left. \begin{aligned} Iwc(\alpha, \beta) &= \int_{\alpha}^{\beta} \cos x_k Riw(a_k, x_k) dx_k \\ Iws(\alpha, \beta) &= \int_{\alpha}^{\beta} \sin x_k Riw(a_k, x_k) dx_k \end{aligned} \right\}. \quad (105)$$

Легко заметить, что функция $Riw(a_k, x_k)$ входит во все соотношения (93–102) как в явном виде, так и в качестве множителя подынтегральных выражений определенных интегралов, превращая их таким образом в двойные интегралы. Поскольку численное интегрирование двойных интегралов требует существенно больших вычислительных ресурсов, преобразование их к простым является весьма полезным при разработке математических моделей, пригодных для решения оптимизационных задач.

Преобразование функций $Iwc(\alpha, \beta)$ и $Iws(\alpha, \beta)$ к простым определенным интегралам

Из соотношений (104) видно, что нижний предел в интегралах (105) может принимать четыре значения: l_{k2} , $2l_{k2}$, $L_k + l_{k2}$ и 0. В использованных ранее при разработке математической модели НВВТП методе наводимых ЭДС и методе среднего потенциала [7, 8] предполагалось, что длины всех проводников во много раз больше их радиусов. Придерживаясь и далее принятого ограничения, можно утверждать, что в интегралах с первыми тремя значениями нижнего предела функцию $Riw(a_k, x_k)$ можно задавать соотношением (64). Подставив (64) в (105) и выполнив интегрирование по dx_k , получим (106).

$$Iwc(\alpha, \beta) = \frac{1}{2} [Si(2\beta) - Si(2\alpha)] - \frac{1}{a_k} \int_0^{a_k} \Phi_{1c}(\alpha, \beta, t) dt - F_c(\alpha, \beta), \quad (106)$$

$$Iws(\alpha, \beta) = \frac{1}{2} \left[\ln \frac{\alpha}{\beta} - Ci(2\beta) - Ci(2\alpha) \right] - \frac{1}{a_k} \int_0^{a_k} \Phi_{1s}(\alpha, \beta, t) dt - F_s(\alpha, \beta), \quad (107)$$

$$Si(x) = \int_0^x \frac{\sin t}{t} dt = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{(2n+1)(2n+1)!}$$

$Ci(x)$ – интегральный косинус:

$$Ci(x) = \gamma + \ln x - \int_0^x \frac{1 - \cos t}{t} dt = \gamma + \ln x + \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n}}{2n(2n)!}$$

$\gamma = 0,5772$ – постоянная Эйлера – Маскерони:

$$\Phi_{1q}(\alpha, \beta, t) = \frac{t}{F_1^2} [TSq(\alpha, \beta, t) + t_{ma} Cq(\alpha, \beta, t)], \quad (108)$$

$q = c, s$ – обобщенный индекс,

$$F_q(\alpha, \beta) = \frac{1}{a_k} \begin{cases} \int_{2,4}^{0,24} \Phi_2^{(0)}(a_k, 0; t) Eq(\alpha, \beta, t) dt, & \alpha > 20a_k \\ \int_{a_k}^{2,4} \Phi_2(a_k, 0; t) Eq(\alpha, \beta, t) dt, & \alpha > 2a_k \end{cases}, \quad (109)$$

$$\begin{cases} Sc(\alpha, \beta, t) \\ Ss(\alpha, \beta, t) \end{cases} = 2 \int_{\alpha}^{\beta} \sin(x_k t_{ma} + \Delta\phi) \begin{cases} \cos x_k \\ \sin x_k \end{cases} dx_k = \begin{cases} Ac(t_{ma}^+; \Delta\phi) + Ac(t_{ma}^-; \Delta\phi) \\ As(t_{ma}^-; \Delta\phi) - As(t_{ma}^+; \Delta\phi) \end{cases}, \quad (110)$$

$$\begin{cases} Cc(\alpha, \beta, t) \\ Cs(\alpha, \beta, t) \end{cases} = 2 \int_{\alpha}^{\beta} \cos(x_k t_{ma}) \begin{cases} \cos x_k \\ \sin x_k \end{cases} dx_k = \begin{cases} As(t_{ma}^-; 0) + As(t_{ma}^+; 0) \\ Ac(t_{ma}^+; 0) - Ac(t_{ma}^-; 0) \end{cases}, \quad (111)$$

$$\begin{cases} Ec(\alpha, \beta, t) \\ Es(\alpha, \beta, t) \end{cases} = 2 \int_{\alpha}^{\beta} \exp(-x_k t_a) \begin{cases} \cos x_k \\ \sin x_k \end{cases} dx_k = \quad (112)$$

$$= \pm \frac{2}{t_a^2 + 1} \left[\exp(-t_a \beta) \begin{cases} \sin \beta - t_a \cos \beta \\ \cos \beta + t_a \sin \beta \end{cases} - \exp(-t_a \alpha) \begin{cases} \sin \alpha - t_a \cos \alpha \\ \cos \alpha + t_a \sin \alpha \end{cases} \right],$$

$$\begin{cases} Ac(b; \Delta\phi) \\ As(b; \Delta\phi) \end{cases} = \frac{1}{b} \begin{cases} \cos(b\alpha + \Delta\phi) - \cos(b\beta + \Delta\phi) \\ \sin(b\beta + \Delta\phi) - \sin(b\alpha + \Delta\phi) \end{cases}, \quad (113)$$

$$t_{ma} = \frac{\sqrt{a_k^2 - t^2}}{a_k}, \quad t_a = \frac{\sqrt{t^2 - a_k^2}}{a_k}, \quad t_{ma}^+ = \frac{\sqrt{a_k^2 - t^2} + a_k}{a_k}, \quad t_{ma}^- = \frac{\sqrt{a_k^2 - t^2} - a_k}{a_k}.$$

При интегрировании (108–111) использованы известные формулы произведений двух гармонических функций. Определенные интегралы (112, 113) получены из неопределенных, приведенных в [13]:

$$\int e^{-ax} \cos x dx = \frac{1}{a^2 + 1} e^{-ax} (\sin x - a \cos x),$$

$$\int e^{-ax} \sin x dx = -\frac{1}{a^2 + 1} e^{-ax} (\cos x - a \sin x).$$

Легко заметить, что в (104) содержатся интегралы $Iwc(0, l_{k2})$ и $Iws(0, l_{k2})$, в которых нижний предел интегрирования равен нулю. В этом случае одним из сомножителей подынтегрального выражения (105) будет $Riw(a_k, 0)$, для описания которого следует воспользоваться соотношениями (40, 43 и 65). Получим (114, 115).

Из соотношений (112) при $\alpha = 0$ учитывая, что $t > 1$, а $a_k < 1$, получим (116, 117).

$$Iwc(0, l_{k2}) = \frac{1}{2} Si(2l_{k2}) - \frac{2}{a_k} \int_0^{a_k} \Phi_{1c}(0, l_{k2}, t) dt - \frac{1}{a_k} \left[\int_{a_k}^{t_{mc}} \Phi_{2c}(t) dt + \int_{t_{mc}}^{\infty} \Phi_{2c}^{(a)}(t) dt \right], \quad (114)$$

$$Iws(0, l_{k2}) = \frac{1}{2} [\gamma + \ln(2l_{k2}) - Ci(2l_{k2})] - \frac{2}{a_k} \int_0^{a_k} \Phi_{1s}(0, l_{k2}, t) dt - \frac{1}{a_k} \left[\int_{a_k}^{t_{ms}} \Phi_{2s}(t) dt + \int_{t_{ms}}^{\infty} \Phi_{2s}^{(a)}(t) dt \right], \quad (115)$$

где

$$\Phi_{2q}(t) = \Phi_2(a_k, 0; t) Eq(0, l_{k2}, t), \quad \Phi_{2q}^{(a)}(t) = \Phi_2^{(a)}(a_k, 0; t) Eq(0, l_{k2}, t).$$

Можно видеть, что если $t > 1$, а $a_k < 1$, то $t_a \approx \frac{t}{a_k}$. Тогда из соотношений (112) при $\alpha = 0$, получим:

$$Ec(0, l_{k2}, t) \approx \frac{2a_k^2}{t^2 + a_k^2} \left[\exp\left(-t \frac{l_2}{a}\right) \times \left(\sin l_{k2} - \frac{t}{a_k} \cos l_{k2}\right) + \frac{t}{a_k} \right]. \quad (116)$$

$$Es(0, l_{k2}, t) \approx \frac{2a_k^2}{t^2 + a_k^2} \left[1 - \exp\left(-t \frac{l_2}{a}\right) \times \left(\cos l_{k2} - \frac{t}{a_k} \sin l_{k2}\right) \right]. \quad (117)$$

Поскольку $l_2/a \gg 1$, соотношения (116 и 117) при $t > 1$ преобразуются к виду:

$$Ec(0, l_{k2}, t) \approx 2a_k/t, \quad (118)$$

$$Es(0, l_{k2}, t) \approx 2a_k^2/t^2. \quad (119)$$

Соотношения (118 и 119) совместно с (68) позволяют представить несобственные интегралы в (114 и 115) следующим образом:

$$\int_{t_{mc}}^{\infty} \Phi_{2c}^{(a)}(t) dt \approx \frac{2Qa_k^2}{\sqrt{\pi}} \int_{t_{mc}}^{\infty} \frac{\cos t + \sin t}{t\sqrt{t}} dt, \quad (120)$$

$$\int_{t_{ms}}^{\infty} \Phi_{2s}^{(a)}(t) dt \approx \frac{2Qa_k^2}{\sqrt{\pi}} \int_{t_{ms}}^{\infty} \frac{\cos t + \sin t}{t^2\sqrt{t}} dt. \quad (121)$$

Для определения рациональных значений t_{mc} и t_{ms} в соотношениях (114 и 115), соответственно, исследованы среднеквадратические отклонения асимптотических приближений функций $\Phi_{2c}(t)$ и $\Phi_{2s}(t)$ на интервале 2π от своих точных значений, выполненных по формуле, аналогичной (74):

$$S_{2q}(t_m) = \sqrt{\frac{1}{2\pi} \int_{t_m}^{t_m+2\pi} \left[\frac{\Phi_{2q}(a_{ki}, 0; t) - \Phi_{2q}^{(a)}(a_{ki}, 0; t)}{M} \right]^2 dt},$$

где $q = c, s$.

Результаты вычислений представлены на рисунке 4. Видно, что для обеспечения среднеквадратического отклонения 0,01 следует выбрать $t_{mc} = t_{ms} = t_m \geq 6$.

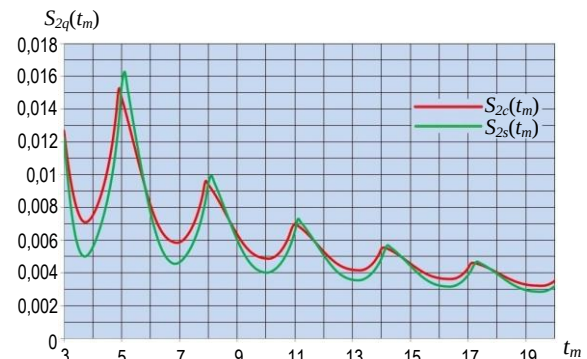


Рис. 4. Среднеквадратическое отклонение функций $\Phi_{2c}^{(a)}(t)/M$ и $\Phi_{2s}^{(a)}(t)/M$ от своих точных значений $\Phi_{2c}(t)/M$ и $\Phi_{2s}(t)/M$ на интервале $t_m \leq t \leq t_m + 2\pi$

Fig. 4. Standard Deviation of Functions $\Phi_{2c}^{(a)}(t)/M$ and $\Phi_{2s}^{(a)}(t)/M$ from Their Exact Values $\Phi_{2c}(t)/M$ and $\Phi_{2s}(t)/M$ on the Interval $t_m \leq t \leq t_m + 2\pi$

Соотношение (120) можно представить, как сумму двух несобственных интегралов, вычислить

которые можно, применив к каждому из них правило интегрирования по частям и воспользовавшись затем соотношением (70) и определением интегралов Френеля (72):

$$\int_{t_m}^{\infty} \frac{\cos t}{t\sqrt{t}} dt = 2 \frac{\cos t_m}{\sqrt{t_m}} - \sqrt{2\pi}[1 - 2S(t_m)], \quad (122)$$

$$\int_{t_m}^{\infty} \frac{\sin t}{t\sqrt{t}} dt = 2 \frac{\sin t_m}{\sqrt{t_m}} + \sqrt{2\pi}[1 - 2C(t_m)]. \quad (123)$$

Аналогичным образом можно представить соотношение (121). Полученные несобственные интегралы также могут быть выражены через интегралы Френеля (124, 125). Для этого к ним следует применить правило интегрирования по частям дважды, а затем также соотношения (70 и 72).

Асимптотические приближения интегралов Френеля для больших аргументов имеют вид (126, 127) [12]. Выражение (126, 127) позволяют найти асимптотические приближения для интегралов (122–125). Получим (128, 129).

Выбирая $t_m = 6$ и ограничиваясь в (128) и (129) первым членом ряда, преобразуем (120) и (121) к следующему виду:

$$\int_{t_{mc}}^{\infty} \Phi_{2c}^{(a)}(t) dt \approx 6,55 \times 10^{-2} Q a_k,$$

$$\int_{t_{ms}}^{\infty} \Phi_{2s}^{(a)}(t) dt \approx 1,95 \times 10^{-2} Q a_k^2.$$

Подставляя эти значения в (114 и 115), получим (130–131).

$$\int_{t_m}^{\infty} \frac{\cos t}{t^2\sqrt{t}} dt = \frac{2}{3} \frac{\cos t_m}{t_m\sqrt{t_m}} - \frac{4}{3} \frac{\sin t_m}{\sqrt{t_m}} - \frac{2\sqrt{2\pi}}{3} [1 - 2C(t_m)], \quad (124)$$

$$\int_{t_m}^{\infty} \frac{\sin t}{t^2\sqrt{t}} dt = \frac{2}{3} \frac{\sin t_m}{t_m\sqrt{t_m}} + \frac{4}{3} \frac{\cos t_m}{\sqrt{t_m}} - \frac{2\sqrt{2\pi}}{3} [1 - 2S(t_m)]. \quad (125)$$

$$C(x) \approx \frac{1}{2} + \frac{1}{\sqrt{2\pi}x} \sum_{n=0}^{[x/2]} (-1)^n \frac{(4n-1)!!}{(2x)^{2n}} \left[\sin x - \frac{(4n+1)}{2x} \cos x \right], \quad (126)$$

$$S(x) \approx \frac{1}{2} - \frac{1}{\sqrt{2\pi}x} \sum_{n=0}^{[x/2]} (-1)^n \frac{(4n-1)!!}{(2x)^{2n}} \left[\cos x + \frac{(4n+1)}{2x} \sin x \right], \quad (127)$$

где $[a]$ – целая часть числа a [13]; $(2n-1)!! = 1 \cdot 3 \cdot 5 \cdot \dots \cdot (2n-1)$, причем $(-1)!! = 1$.

$$\left. \begin{aligned} \int_{t_m}^{\infty} \frac{\cos t}{t\sqrt{t}} dt &\approx -\frac{\sin t_m}{t_m\sqrt{t_m}} + \frac{2}{\sqrt{t_m}} \sum_{n=1}^{[t_m/2]} (-1)^n \frac{(4n-1)!!}{(2t_m)^{2n}} \left[\cos t_m + \frac{(4n+1)}{2t_m} \sin t_m \right] \\ \int_{t_m}^{\infty} \frac{\sin t}{t\sqrt{t}} dt &\approx \frac{\cos t_m}{t_m\sqrt{t_m}} - \frac{2}{\sqrt{t_m}} \sum_{n=1}^{[t_m/2]} (-1)^n \frac{(4n-1)!!}{(2t_m)^{2n}} \left[\sin t_m - \frac{(4n+1)}{2t_m} \cos t_m \right] \end{aligned} \right\} \quad (128)$$

$$\left. \begin{aligned} \int_{t_m}^{\infty} \frac{\cos t}{t^2\sqrt{t}} dt &\approx \frac{4}{3\sqrt{t_m}} \sum_{n=1}^{[t_m/2]} (-1)^n \frac{(4n-1)!!}{(2t_m)^{2n}} \left[\sin t_m - \frac{(4n+1)}{2t_m} \cos t_m \right] \\ \int_{t_m}^{\infty} \frac{\sin t}{t^2\sqrt{t}} dt &\approx -\frac{4}{3\sqrt{t_m}} \sum_{n=1}^{[t_m/2]} (-1)^n \frac{(4n-1)!!}{(2t_m)^{2n}} \left[\cos t_m + \frac{(4n+1)}{2t_m} \sin t_m \right] \end{aligned} \right\}. \quad (129)$$

$$Iwc(0, l_{k2}) = \frac{1}{2} Si(2l_{k2}) - \frac{2}{a_k} \int_0^{a_k} \Phi_{1c}(0, l_{k2}, t) dt - \frac{1}{a_k} \int_{a_k}^6 \Phi_{2c}(t) dt - 6,55 \times 10^{-2} Q, \quad (130)$$

$$Iws(0, l_{k2}) = \frac{1}{2} [\gamma + \ln(2l_{k2}) - Ci(2l_{k2})] - \frac{2}{a_k} \int_0^{a_k} \Phi_{1s}(0, l_{k2}, t) dt - \frac{1}{a_k} \int_{a_k}^6 \Phi_{2s}(t) dt - 1,95 \times 10^{-2} Q a_k. \quad (131)$$

Формулы (130 и 131) совместно с (106 и 107) преобразуют двойные интегралы (105) в простые, что позволяет существенно сократить время вычислений соотношений (104). Учитывая, что (64 и 74)

также делают пригодными для инженерных расчетов соотношения (103), вычисление взаимных сопротивлений нагрузочного восьмиполюсника БТ становится простой инженерной задачей.

Входное сопротивление, коэффициент полезного действия и коэффициент усиления НВВТП

Входное сопротивление НВВТП, как следует из эквивалентной схемы (см. рисунок 1), будет:

$$Z_{\text{вх}} = R_{\Sigma} + R_{\text{пот}} + iX, \quad (132)$$

где $R_{\Sigma} = \sum_p \sum_{p'} n_p^* n_{p'} r_{pp'}$ – сопротивление излучения (133); $R_{\text{пот}} = - (I_m(\tilde{C})) / (\omega |\tilde{C}|^2)$ – сопротивление потерь в подстилающей поверхности (134); $X = - (\text{Re}(\tilde{C})) / (\omega |\tilde{C}|^2)$ – емкостное сопротивление НВВТП (135).

Как видно из (133), сопротивление излучения НВВТП зависит не только от параметров восьмиполосника внутренних сопротивлений, а и от КТТЗА к ПБТ. В первой части этой работы [1] получены аналитические выражения, устанавливающие связь этих величин с геометрическими размерами и комплексной емкостью НВВТП:

$$|n_{c1}| = \frac{\sin l_{k1}}{\rho_1} \sqrt{(R_c - R_c^0)^2 + (X_c - X_c^0)^2}, \quad (136)$$

$$\phi_{c1} = \arctg[(R_c^0 - R_c) / (X_c - X_c^0)], \quad (137)$$

$$|n_{c2}| = \frac{\sin l_{k2}}{\rho_2} \sqrt{(\rho_1 \text{ctgl}_{k1} + \rho_2 \text{ctgl}_{k2} + X_c^0)^2 + (R_c^0)^2}, \quad (138)$$

$$\phi_{c2} = \arctg[-R_c^0 / (\rho_1 \text{ctgl}_{k1} + \rho_2 \text{ctgl}_{k2} + X_c^0)], \quad (139)$$

$$n_{c1} = (1 - n_{ci} \cos l_{ki}) / \sin l_{ki}, \quad (140)$$

$$\rho_1 = 60 \left(\ln \frac{l_{k1}}{a_{k1}} - 1 - \frac{L_k + l_{k2}}{l_{k2}} \ln \frac{L_k}{L_k + l_{k2}} \right) \quad (141)$$

$$\rho_2 = 60 \left\{ \ln \frac{l_{k1}}{a_{k1}} - 1 - \frac{1}{2} \left[\left(1 - \frac{l_{k2}}{l_{k1}} \right) \ln \frac{L_k}{L_k + l_{k2}} - \frac{l_{k2}}{l_{k1}} \ln \frac{4l_{k2}}{L_k + l_{k2}} \right] \right\}, \quad (142)$$

$$R_c^0 + iX_c^0 = 1/i\omega \tilde{C}_0, \quad (143)$$

$$R_c + iX_c = 1/i\omega \tilde{C}, \quad (144)$$

где $\tilde{C}_0$ – комплексная емкость НВВТП без верхней нагрузки; $\tilde{C}$ – комплексная емкость НВВТП.

Соотношения, определяющие комплексную емкость НВВТП произвольного вида, а также сопротивление тепловых потерь в подстилающей поверхности и реактивную составляющую входного сопротивления, получены во второй части этой работы [8]. Раздельный учет сопротивления потерь и сопротивления излучения позволяет легко определить КПД НВВТП относительно поглощаемой мощности [2, 3]:

$$\eta = R_{\Sigma} / (R_{\Sigma} + R_{\text{пот}}). \quad (145)$$

Коэффициент согласования по сопротивлению НВВТП с 50-омным фидерным трактом [2, 3, 6]:

$$\eta_{\text{сг}} = \frac{4 \cdot 50 \cdot (R_{\Sigma} + R_{\text{пот}})}{(R_{\Sigma} + R_{\text{пот}} + 50)^2 + X^2}. \quad (146)$$

Соотношения (145 и 146) позволяют определить КПД НВВТП относительно подводимой мощности [2]:

$$\eta_{\Sigma} = \eta_{\text{сг}} = \frac{200 \cdot R_{\Sigma}}{(R_{\Sigma} + R_{\text{пот}} + 50)^2 + X^2}. \quad (147)$$

Поскольку высота излучателя (суммарная высота вертикальных проводников) НВВТП ВЧ-диапазона, как правило, существенно меньше длины волны, его коэффициент направленного действия можно считать равным 3 [2, 3, 15 и др.]. Следовательно, КУ НВВТП полностью определяется его КПД и равен [2, 3, 6]:

$$G = 3\eta_{\Sigma}. \quad (148)$$

Таким образом, полученные в работе соотношения устанавливают связь основных электрических характеристик НВВТП с его геометрическими размерами и проводимостью подстилающей поверхности. Это позволяет считать представленную совокупность формул математической моделью НВВТП.

Экспериментальная проверка полученных результатов и рекомендации по повышению КУ ШТ4Н81

Для проверки адекватности разработанной математической модели было осуществлено сравнение расчетной и экспериментально полученной зависимости КУ НВВТП ШТ4Н81 от частоты. Основные размеры антенны ШТ4Н81: высота $L = 2$ м; верхнее плечо излучателя $l_1 = 1$ м; нижнее плечо излучателя $l_2 = 1$ м; длина заземлителя $l_3 = 0,4$ м; радиусы проводников излучателя и заземлителя $a = 0,025$ м. Верхней оконечной нагрузки и противовесов исследуемая антенна не содержит.

Экспериментальное определение КУ осуществлялось путем измерения напряженности электрического поля, создаваемого ШТ4Н81 на удалении $r = 140$ м, при подводимой к антенне мощности $P = 5 \cdot 10^{-3}$ Вт.

Как известно [3], напряженность поля ЭМВ с вертикальной поляризацией, распространяющейся вдоль плоской полупроводящей поверхности, определяется по формуле Шулейкина – Ван дер Поля:

$$E = \frac{\sqrt{30PG}}{r} |F|, \quad (149)$$

где

$$|F| = \frac{2 + 0,3x}{2 + x + 0,6x^2} - \sqrt{\frac{x}{2}} e^{-\frac{5x}{8}} \frac{\varepsilon}{\sqrt{\varepsilon^2 + (60\sigma\lambda)^2}} \quad (150)$$

$$x = \frac{\pi r}{\lambda} \frac{1}{\sqrt{\varepsilon^2 + (60\sigma\lambda)^2}}, \quad (151)$$

$|F|$ – модуль множителя ослабления трассы; x – приведенное расстояние.

Из соотношения (149) следует:

$$G = \frac{E^2 r^2}{30P|F|^2}. \quad (152)$$

Измерение напряженности электрического поля проводилась на частотах от 1,5 до 15 МГц, после чего все величины, входящие в правую часть соотношения (152), становились известными, за исключением параметров подстилающей поверхности. Задавая одинаковые параметры почвы в математической модели ШТ4Н81 и в соотношении (152), было установлено, что частотные зависимости расчетного значения КУ и КУ, полученного из (152), наилучшим образом совпадают при $\varepsilon = 4$ и $\sigma = 0,04$ (см/м). На рисунке 5 представлены эти зависимости.

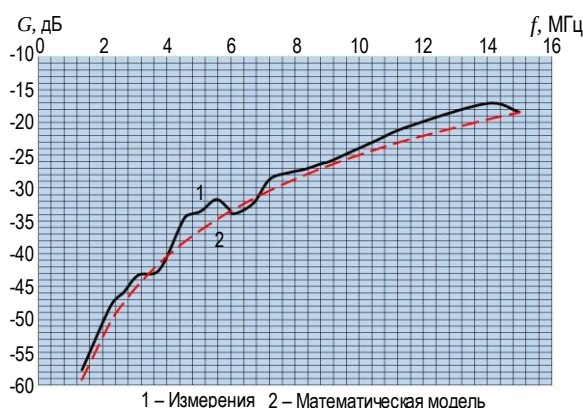


Рис. 5. Частотная зависимость КУ НВВТП ШТ4Н81

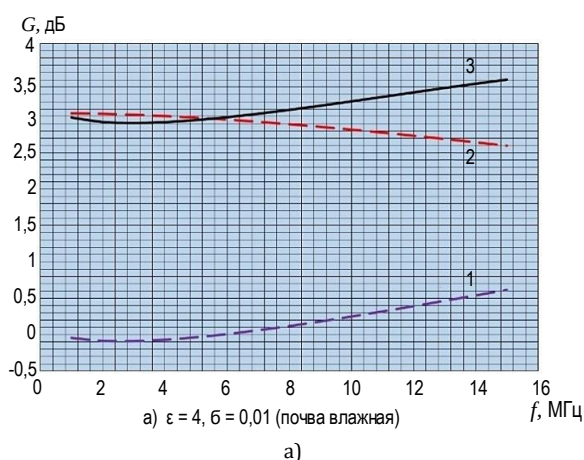
Fig. 5. Frequency Dependence of Gain USMF ShT4N81

Легко заметить, что их расхождение в большей части исследуемого диапазона лежит в пределах точности измерения напряженности поля. Такое

совпадение свидетельствует о возможности применения представленной модели для оценки оптимальности конструкции ШТ4Н81. КУ существенно меньше единицы, что объясняется малыми электрическими размерами антенны и установкой ее непосредственно на полупроводящей поверхности. Поскольку конструкция ШТ4Н81, с целью обеспечения минимального времени разворачивания, выбрана предельно простой, целесообразно оценить возможность повышения КУ включением в ее состав таких дополнительных элементов, как противовесы и верхняя оконечная нагрузка.

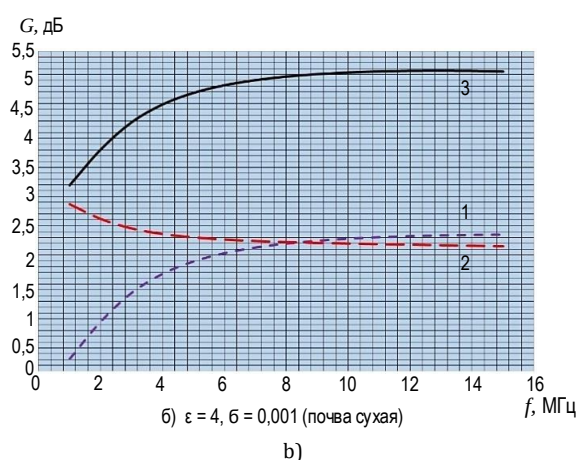
На рисунке 6 представлены частотные зависимости ожидаемого прироста КУ от введения каждого из этих элементов и от их совместного действия. Расчеты сделаны как для влажной, так и для сухой почвы. Верхняя оконечная нагрузка предполагалась состоящей из восьми, радиально расходящихся в горизонтальной плоскости проводников, длиной по 0,5 м. Система противовесов также состояла из восьми, радиально расходящихся по поверхности почвы, проводников по 2 м.

Легко заметить, что эффект от противовесов на влажной почве практически отсутствует, что было подтверждено экспериментально. В процессе измерения напряженности поля на удалении 140 м от ШТ4Н81, подключение противовесов никак не сказалось на уровне сигнала на входе измерительного приемника. Ожидаемый эффект от подключения нагрузки (см. рисунок 6) составляет $(2 \div 3)$ дБ. Совместное использование противовесов и верхней нагрузки позволяет рассчитывать на увеличение КУ на сухой почве на $(3 \div 5)$ дБ. Однако следует учесть, что подобная доработка антенны приведет к заметному увеличению времени ее разворачивания, поэтому решение на доработку должно приниматься только после анализа особенностей эксплуатации комплекса, в которой данная антенна входит.



а) $\varepsilon = 4, \sigma = 0,01$ (почва влажная)

а)



б) $\varepsilon = 4, \sigma = 0,001$ (почва сухая)

б)

Рис. 6. Частотная зависимость прироста КУ от введения в конструкцию НВВТП ШТ4Н81 противовесов (1), верхней нагрузки (2), а также противовесов и верхней нагрузки (3): а) почва влажная; б) почва сухая

Fig. 6. Frequency Dependence of the Gain in Gain on the Introduction of USMF into the Design ShT4N81: 1 – Counterweights; 2 – Top Load; 3 – Counterweights and Top Load (3): а) the Wet Soil; б) the Dry Soil

Заключение

Соотношения (75, 80–89, 93–102, 132–148) совместно с описанными в [8] выражениями, определяющие комплексную емкость НВВТП, представляют собой математическую модель электрически малого НВВТП ВЧ-диапазона, развернутого на полупроводящей поверхности и предназначенного для работы поверхностной волной. Данная модель связывает важнейшие характеристики НВВТП с его геометрическими размерами и электрическими параметрами почвы. Все входящие в математическую модель соотношения доведены до аналитического вида, за исключением нескольких определенных интегралов, вычисление которых не требует сколько-нибудь серьезного вычислительного ресурса и может выполняться с помощью среднего персонального компьютера. Экспериментальные исследования, выполненные в рамках этой работы, подтвердили связь как коэффициента усиления антенны, так и множителя ослабления трассы с электрическими характеристиками подстилающей поверхности. Показано, что подбором двух параметров комплексной относительной диэлектрической проницаемости земли можно обеспечить совпадение измеренной и расчетной напряженности поля

на заданной трассе во всем исследуемом диапазоне частот с точностью до погрешности измерения. Этот факт является свидетельством адекватности предлагаемой модели и позволяет рассчитывать на ее полезность при решении большого числа прикладных задач. К числу таких задач можно отнести частотно-территориальное планирование ВЧ радиосвязи, планирование мероприятий по местопредопределению несанкционированных источников излучения, оптимизацию конструкции НВВТП и некоторые другие.

Однако, как известно [4, 6], параметры подстилающей поверхности не являются стабильной величиной и могут существенно изменяться в зависимости от интенсивности осадков и времени года. В связи с этим предлагаемая математическая модель может оказаться полезной для оперативной коррекции их усредненных значений. Коррекция параметров трассы в реальном масштабе времени путем контроля уровня тестовых сигналов позволит повысить точность определения радиусов зон электромагнитной доступности источников ЭМВ, с учетом их энергетического потенциала и расположения на местности.

Список источников

1. Попов О.В., Тумашов А.В., Борисов Г.Н., Коровин К.О. Методика расчета радиуса зоны электромагнитной доступности источника поверхностных волн с заданной плотностью спектра изотропно излучаемого сигнала // Труды учебных заведений связи. 2022. Т. 8. № 3. С. 72–79. DOI:10.31854/1813-324X-2022-8-3-72-79
2. Гавеля Н.П., Истрашкин А.Д., Муравьев Ю.К., Серков В.П. Антенны. Ч. I. Л.: ВКАС, 1963. 633 с.
3. Серков В.П. Распространение радиоволн и антенные устройства. Л.: ВАС, 1981.
4. Калинин А.И., Черенкова Е.Л. Распространение радиоволн и работа радиолиний. М.: Связь, 1971. 439 с.
5. Бесчастнов Н.С., Конторович М.И. О потерях в земле при использовании корпуса передатчика в качестве противовеса // Труды ВКАС. 1944. № 3.
6. Муравьев Ю.К. Справочник по расчету проволочных антенн. Л.: ВАС, 1978.
7. Попов О.В., Тумашов А.В., Борисов Г.Н., Коровин К.О. Математическая модель несимметричного вибратора с вынесенной точкой питания. Часть 1. Общий подход к построению математической модели // Труды учебных заведений связи. 2023. Т. 9. № 1. С. 24–33. DOI:10.31854/1813-324X-2023-9-1-24-33
8. Попов О.В., Тумашов А.В., Борисов Г.Н., Коровин К.О. Математическая модель несимметричного вибратора с вынесенной точкой питания. Часть 2. Определение комплексной емкости малых, по сравнению с длиной волны, нормально разомкнутых проволочных антенн // Труды учебных заведений связи. 2023. Т. 9. № 2. С. 6–21. DOI:10.31854/1813-324X-2023-9-2-6-21
9. Стрэттон Дж.А. Теория электромагнетизма. Пер. с англ. М.; Л.: Гостехиздат, 1948. 539 с.
10. Корн Г., Корн Т. Справочник по математике для научных работников и инженеров. Пер. с англ. СПб.: Лань, 2003. 831 с.
11. Смирнов В.И. Курс высшей математики. М.: Наука, 1974. Т. 1.
12. Янке Е., Эмде Ф., Леш Ф. Специальные функции. Пер. с нем. М.: Наука, 1977. 342 с.
13. Прудников А.П., Брычков Ю.А., Маричев О.И. Интегралы и ряды. Элементарные функции. М.: Наука. Главная редакция физико-математической литературы, 1981.
14. Анго А. Математика для электро- и радиоинженеров. М.: Наука, 1967.
15. Вершков М.В. Судовые антенны. Л.: Судостроение, 1972. 424 с.

References

1. Popov O., Tumashov A., Borisov G., Korovin K. Method for Calculating the Radius of an Electromagnetic Availability Zone based on a Surface Wave Source with a Given Spectrum Density of an Isotropically Emitted Signal. *Proceedings of Telecommun. Univ.* 2022;8(3):72–79. DOI:10.31854/1813-324X-2022-8-3-72-79
2. Gavelya N.P., Istrashkin A.D., Muravyov Yu.K. *Antennas. Part I.* Leningrad: Military Academy of Communications Publ.; 1963. 633 p.
3. Serkov V.P. *Radio Wave Propagation and Antenna Devices.* Leningrad: Military Academy of Telecommunications Publ.; 1981.

4. Kalinin A.I., Cherenkova E.L. *Propagation of Radio Waves and the Operation of Radio Links*. Moscow: Sviaz Publ.; 1971. 439 p.
5. Beschastnov N.S., Kontorovich M.I. On Loss in the Ground when Assembling the Mechanism as a Counterweight. *Trudy VKAS*. 1944;3.
6. Muravyov Yu.K. *Handbook for the Calculation of Wire Antennas*. Leningrad: Military Academy of Telecommunications Publ.; 1978.
7. Popov O., Tumashov A., Borisov G., Korovin K. Mathematical Model of the Unbalanced Monopole Feed. Part 1. General Approach to Building a Mathematical Model. *Proceedings of Telecommun. Univ.* 2023;9(1):24–33. DOI:10.31854/1813-324X-2023-9-1-24-33
8. Popov O., Tumashov A., Borisov G., Korovin K. Mathematical Model of the Unbalanced Monopole Feed. Part 2. Determination of Complex Capacitance of Normally Open Wire Antennas, Small with Respect to the Wavelength. *Proceedings of Telecommun. Univ.* 2023;9(2):6–21. DOI:10.31854/1813-324X-2023-9-2-6-21
9. Stratton J. *Electromagnetic Theory*. NY: McGraw – Hill; 1941.
10. Korn A., Korn T.M. *Mathematical Handbook for Scientists and Engineers. Definitions, Theorems, and Formulas for Reference and Review*. NY: McGraw – Hill; 1961. 943 p.
11. Smirnov V.I. *Higher Mathematics Course, vol.1*. Moscow: Nauka Publ.; 1974.
12. Janke, Emde, Lösch. *Tafel Höherer Funktionen*. Stuttgart; 1960. DOI:10.1002/zamm.19610410619
13. Prudnikov A.P., Brychkov Yu.A., Marichev O.I. *Integrals and Series. Elementary Functions*. Moscow: Nauka Glavnaia redaktsiia fiziko-matematicheskoi literatury Publ.; 1981.
14. Ango A. *Mathematics for Electrical and Radio Engineers*. Moscow: Nauka Publ.; 1967.
15. Vershkov M.V. *Ship Antennas*. Leningrad: Sudostroenie Publ.; 1972. 424 p.

Статья поступила в редакцию 08.11.2023; одобрена после рецензирования 13.11.2023; принята к публикации 22.11.2023.

The article was submitted 08.11.2023; approved after reviewing 13.11.2023; accepted for publication 22.11.2023.

Информация об авторах:

ПОПОВ
Олег Вениаминович

кандидат технических наук, доцент, ведущий научный сотрудник ООО «Специальный технологический центр»

 <https://orcid.org/0000-0002-5315-2679>

ТУМАШОВ
Андрей Витальевич

инженер-конструктор ООО «Специальный технологический центр»

 <https://orcid.org/0000-0003-2656-0463>

БОРИСОВ
Георгий Николаевич

инженер ООО «Специальный технологический центр»

 <https://orcid.org/0000-0002-3275-251X>

КОРОВИН
Константин Олегович

кандидат физико-математических наук, доцент, заведующий кафедрой радиосистем и обработки сигналов Санкт-Петербургского государственного университета телекоммуникаций им. проф. М.А. Бонч-Бруевича

 <https://orcid.org/0000-0001-7979-3725>

ЭЛЕКТРОНИКА, ФОТОНИКА, ПРИБОРОСТРОЕНИЕ И СВЯЗЬ

**2.2.6 – Оптические
и оптико–электронные приборы
и комплексы**

**2.2.13 – Радиотехника, в том числе системы
и устройства телевидения**

**2.2.14 – Антенны, СВЧ–устройства
и их технологии**

**2.2.15 – Системы, сети и устройства
телекоммуникаций**

2.2.16 – Радиолокация и радионавигация

Научная статья

УДК 621.372.632

DOI:10.31854/1813-324X-2023-9-6-25-33



Границы оценивания параметров сигналов с прямым расширением спектра

Ефим Александрович Брусин, brusin.ea@sut.ru

АО «Российский институт радионавигации и времени»,
Санкт-Петербург, 192012, Российская Федерация
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича,
Санкт-Петербург, 193232, Российская Федерация

Аннотация: В статье обсуждаются проблемы оценивания параметров сигналов с прямым расширением спектра и анализа эффективности получаемых оценок. Достижимая дисперсия оценок, как правило, сопоставляется с соответствующей границей Крамера – Рао. Предложенный подход позволяет определить границы Крамера – Рао для сигналов различных видов модуляции и спектральных характеристик. Представлены зависимости границ оценивания от отношения сигнал/шум, длины расширяющей последовательности и длительности интервала наблюдения.

Ключевые слова: прямое расширение спектра, оценивание параметров сигнала, граница Крамера – Рао

Ссылка для цитирования: Брусин Е.А. Границы оценивания параметров сигналов с прямым расширением спектра // Труды учебных заведений связи. 2023. Т. 9. № 6. С. 25–33. DOI:10.31854/1813-324X-2023-9-6-25-33

Direct Sequence Spread Spectrum Signal's Parameters Estimation Bounds

Ефим Брусин, brusin.ea@sut.ru

Russian Institute of Radionavigation and Time,
St. Petersburg, 192012, Russian Federation
The Bonch-Bruevich Saint-Petersburg State University of Telecommunications,
St. Petersburg, 193232, Russian Federation

Abstract: The article discusses the problems of direct spread spectrum signal's parameters estimation and estimation performance analyzing. The achieved estimation variances compared with the corresponding Cramer – Rao bounds as a rule. The proposed approach makes it possible to determine the bounds for signals of various types of modulation and spectral characteristics. The dependences of the Cramer – Rao bounds on the signal-to-noise ratio, the spread spectrum sequence length and the duration of the observation interval are presented.

Keywords: direct sequence spread spectrum, signal's parameters estimation, Cramer – Rao bound

For citation: Brusin E. Direct Sequence Spread Spectrum Signal's Parameters Estimation Bounds. *Proceedings of Telecommun. Univ.* 2023;9(6):25–33. DOI:10.31854/1813-324X-2023-9-6-25-33

Постановка задачи

Фактически саму демодуляцию можно трактовать как процедуру совместного оценивания параметров принимаемого сигнала. Но, как правило, начальная неопределенность по частоте несущего колебания и задержке таковы, что реализации собственно демодуляции невозможна без предварительного оценивания указанных параметров. В этом смысле оценивание условно разделяется на оценивание параметров до демодуляции, определяемое, как начальная синхронизация, и оценивание в процессе демодуляции.

Проблемы реализации начальной синхронизации, в частности, подробно обсуждались в работах [1–3]. В этих работах основной акцент делается на оценивании несущей частоты, а оценивание задержки фактически сводится к «грубому» определению положения корреляционного пика в одном из частотных каналов. Границы Крамера – Рао для оценивания несущей частоты и задержки приведены, например, в работе [4]. В работе показано, что границы зависят от среднеквадратичной полосы и среднеквадратичной длительности сигнала, но не раскрывается смысл этих параметров. Количественно не определена связь границ оценивания со спектральными характеристиками принимаемых сигналов.

Как правило, оценивание параметров сигналов с прямым расширением спектра сопряжено с анализом приема сигналов с существенным доплеровским смещением несущей частоты. Например, в таких системах, как системы спутниковой навигации. Однако сигналы с прямым расширением спектра используются также в системах, построенных на основе геостационарных спутников [5, 6]. В таких системах особое значение играет оценивание несущей частоты и фазы несущего колебания, так как точность оценивания последнего параметра определяет, например, качество фазовых измерений псевдоудальности.

В то же время в работах по демодуляции сигналов с прямым расширением спектра практически не обсуждаются границы оценивания фазы несущего колебания. В свою очередь, несмотря на наличие большого количества работ по проблемам приема сигналов с прямым расширением спектра, в известных работах не приводятся значения границ оценивания, учитывающих в полной мере специфику используемых сигналов. Поэтому проблема определения указанных границ представляется весьма актуальной.

В этой связи основной задачей представляемой статьи является получение границ оценивания параметров сигналов с прямым расширением спектра для различных видов модуляции и спектральных характеристик принимаемых сигналов.

Границы оценивания

В системах, использующих сигналы с прямым расширением спектра, как правило, применяются фазовые методы модуляции. Передаваемый сигнал формируется путем умножения на расширяющую спектр последовательность $s_k(t)$ (сигнатуру):

$$s_k(t; b_k) = s_k(t) \cdot B_k(t) \cdot \cos(2\pi f_0 t),$$

где $B_k(t) = d_k = \pm 1$ – информационные символы, передаваемые на интервалах T_b ; T_b – длительность информационного символа (бита); f_0 – номинальное значение частоты несущего колебания.

Длительность бита:

$$T_b = N_I T_S,$$

где T_S и N_I – длительность чипа и выраженная в количестве чипов длина расширяющей последовательности, соответственно.

Упрощенная схема формирования сигнала с прямым расширением спектра представлена на рисунке 1 [2].

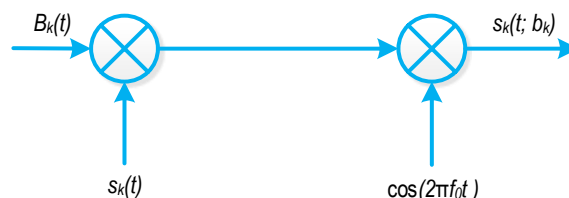


Рис. 1. Формирование сигнала с прямым расширением спектра

Fig. 1. Direct Spread Spectrum Signal Generation

Принимаемый сигнал может быть представлен в виде:

$$\tilde{s}_k(t; b_k) = s_k(t - \hat{\tau}) \cdot B_k(t - \hat{\tau}) \times \cos(2\pi(f_0 + f)t + \varphi) + n_\tau(t),$$

где $\hat{\tau}$ – задержка в канале; f – смещение частоты несущего колебания относительно заданного номинального значения; φ – фаза несущей частоты; $n_\tau(t)$ – отсчеты шума.

Граница Крамера – Рао дисперсии оценки задержки определяется следующим образом [4]:

$$D\{\hat{\tau}\} \approx \frac{1}{4\pi^2 W_{RMS}^2 q^2}, q \gg 1, \quad (1)$$

где q^2 – отношение сигнал-шум; W_{RMS} – среднеквадратичная полоса сигнала.

Квадрат среднеквадратичной полосы сигнала определяется выражением [7]:

$$W_{RMS}^2 = \frac{\int_{-\infty}^{\infty} f^2 |G(f)|^2 df}{\int_{-\infty}^{\infty} |G(f)|^2 df}, \quad (2)$$

где $|G(f)|^2$ – энергетический спектр низкочастотного модулирующего колебания, формирующего передаваемый сигнал.

Для сигнала фазовой модуляции энергетический спектр низкочастотного модулирующего колебания выражается следующим образом [8]:

$$|G(f)|^2 \sim \left(\frac{\sin(2\pi f T_s)}{2\pi f T_s} \right)^2, \quad (3)$$

где T_s – длительность чипа передаваемого сигнала.

Фазовая модуляция наиболее распространена в системах связи и навигации, использующих сигналы с прямым расширением спектра. В главном «лепестке» спектра (3) содержится более 90 % мощности сигнала [8]. При вычислении выражения (2) для фазомодулированного сигнала будем полагать, что сигнал ограничен полосой $\left[-\frac{1}{2T_s}; \frac{1}{2T_s}\right]$.

Для такого ограничения (2) переписывается в виде:

$$W_{PM}^2 = \int_{-T_s/2}^{T_s/2} f^2 \left(\frac{\sin(2\pi f T_s)}{2\pi f T_s} \right)^2 df / \int_{-T_s/2}^{T_s/2} \left(\frac{\sin(2\pi f T_s)}{2\pi f T_s} \right)^2 df,$$

где W_{PM} – среднеквадратичная полоса сигнала фазовой модуляции для сигнала со спектром, определенным (3).

В итоге для W_{PM}^2 получим:

$$W_{PM}^2 = \frac{1}{4\pi T_s^2} / \int_{-\pi}^{\pi} \left(\frac{\sin x}{x} \right)^2 dx. \quad (4)$$

Отношение сигнал/шум в (1) выражается следующим образом:

$$q^2 = 2N_I E_s / N_0,$$

где E_s / N_0 – отношение сигнал/шум на чип.

С учетом (4) получим выражение для минимальной дисперсии оценивания задержки:

$$D_{PM}\{\hat{\tau}\} \approx \frac{T_s^2}{2\pi N_I E_s / N_0} \int_{-\pi}^{\pi} \left(\frac{\sin x}{x} \right)^2 dx.$$

Как правило, в работах по теории оценивания параметров сигналов оперируют с нормированной

границей Крамера – Рао [9].

В рассматриваемом случае для фазовой модуляции нормированная к длительности чипа граница выражается следующим образом:

$$CRB_{PM}(\hat{\tau}) = D_{PM}\{\hat{\tau}\} / T_s^2 \approx \frac{1}{2\pi N_I E_s / N_0} \int_{-\pi}^{\pi} \left(\frac{\sin x}{x} \right)^2 dx. \quad (5)$$

Кроме собственно фазовой модуляции, в системах широко используются модуляция минимального сдвига (ММС) и сигналы, сформированные с использованием фильтра Найквиста.

Энергетический спектр низкочастотного модулирующего колебания для ММС определяется выражением [8]:

$$|G(f)|^2 \sim \frac{1 + \cos(4\pi f T_s)}{(1 - 16f^2 T_s^2)^2}. \quad (6)$$

И, наконец, если импульсная характеристика фильтра, формирующего сигнал, соответствует импульсной характеристике фильтра Найквиста, энергетический спектр соответствующего низкочастотного колебания определяется амплитудно-частотной характеристикой фильтра, которая выражается в виде (7), где α – коэффициент скругления фильтра Найквиста, формирующего сигнал на передачу.

На рисунке 2 представлены нормированные спектры для сигналов фазовой модуляции и сигналов модуляции минимального сдвига. На рисунке 3 – спектры Найквиста для двух коэффициентов скругления. В главном «лепестке» спектра сигнала ММС содержится 97 % мощности сигнала [8]. Соответственно, также как и для ФМ-сигнала, будем рассматривать полосу главного «лепестка» $\left[-\frac{3}{4T_s}; \frac{3}{4T_s}\right]$.

В соответствии с рассматриваемым условием квадрат среднеквадратичной полосы для сигнала ММС определяется формулой (8). Упростив (8), получим окончательное выражение (9) для W_{MSK}^2 .

$$|H(f, \alpha)|^2 = \begin{cases} T_s, & |f| < \frac{1-\alpha}{2T_s} \\ T_s \cos^2 \left[\frac{\pi}{4\alpha} (|2fT_s| - 1 + \alpha) \right], & \frac{1-\alpha}{2T_s} \leq |f| \leq \frac{1+\alpha}{2T_s} \\ 0, & |f| > \frac{1+\alpha}{2T_s} \end{cases} \quad (7)$$

$$W_{MSK}^2 = \int_{-\frac{3}{4T_s}}^{\frac{3}{4T_s}} f^2 \frac{1 + \cos(4\pi f T_s)}{(1 - 16f^2 T_s^2)^2} df / \int_{-\frac{3}{4T_s}}^{\frac{3}{4T_s}} \frac{1 + \cos(4\pi f T_s)}{(1 - 16f^2 T_s^2)^2} df. \quad (8)$$

$$W_{MSK}^2 = \frac{1}{16\pi^2 T_s^2} \int_{-3\pi}^{3\pi} x^2 \frac{1 + \cos x}{(1 - (x/\pi)^2)^2} dx / \int_{-3\pi}^{3\pi} \frac{1 + \cos x}{(1 - (x/\pi)^2)^2} dx. \quad (9)$$

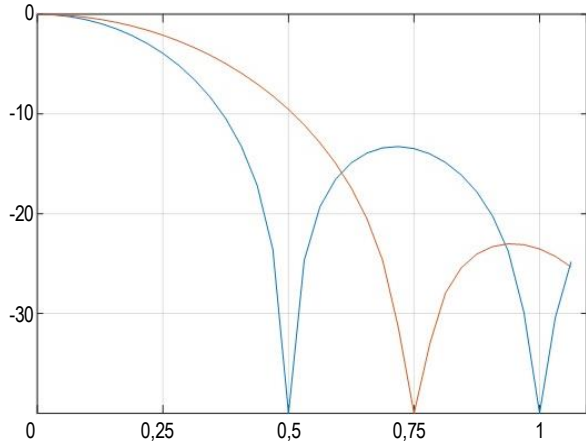


Рис. 2. ФМ и ММС: нормированные энергетические спектры

Fig. 2. PSK and MSK: Normalized Signal Spectrums

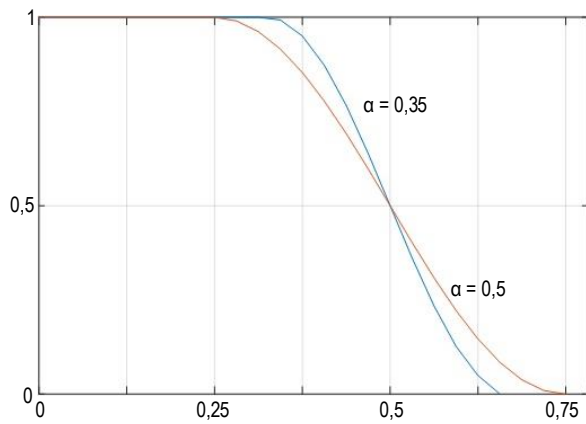


Рис. 3. Спектры Найквиста

Fig. 3. Nyquist Spectrums

Введем обозначение:

$$\xi_{MSK} = \int_{-3\pi}^{3\pi} x^2 \frac{1 + \cos x}{(1 - (x/\pi)^2)^2} dx / \int_{-3\pi}^{3\pi} \frac{1 + \cos x}{(1 - (x/\pi)^2)^2} dx.$$

С учетом (9) граница (1) для ММС переписывается в виде:

$$D_{MSK}\{\hat{\tau}\} \approx \frac{2T_S^2}{\xi_{MSK} N_I E_S / N_0}.$$

В итоге для модуляции минимального сдвига нормированная граница Крамера – Рао оценки задержки определяется выражением:

$$CRB_{MSK}(\hat{\tau}) = D_{MSK}\{\hat{\tau}\} / T_S^2 \approx \frac{2}{\xi_{MSK} N_I E_S / N_0}. \quad (10)$$

И, наконец, квадрат среднеквадратичной полосы сигнала, сформированного спектром Найквиста W_α^2 , определяется выражением [9]:

$$W_\alpha^2 = \frac{1}{T_S^2} \left(\frac{1}{12} + \alpha^2 \left(\frac{1}{4} - \frac{2}{\pi^2} \right) \right). \quad (11)$$

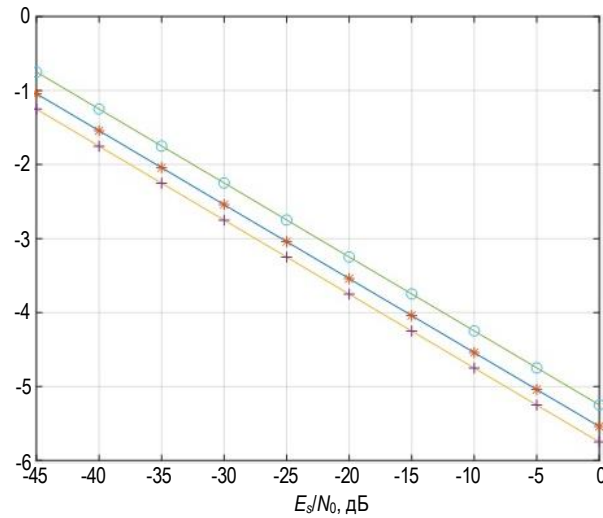
С учетом (11) граница Крамера – Рао оценки задержки для сигнала, сформированного фильтром Найквиста, определяется выражением:

$$D_\alpha\{\hat{\tau}_k\} \approx \frac{T_S^2}{8\pi^2 N_I E_S / N_0 \left(\frac{1}{12} + \alpha^2 \left(\frac{1}{4} - \frac{2}{\pi^2} \right) \right)}.$$

Соответственно, запишем нормированную границу Крамера – Рао оценки задержки для сигнала с «найквистовским» спектром:

$$CRB_\alpha(\hat{\tau}_k) = D_\alpha\{\hat{\tau}_k\} / T_S^2 \approx \frac{1}{8\pi^2 N_I E_S / N_0 \left(\frac{1}{12} + \alpha^2 \left(\frac{1}{4} - \frac{2}{\pi^2} \right) \right)}. \quad (12)$$

На рисунке 4 представлены зависимости границ (5, 10 и 11) от отношения сигнал/шум на чип для длины расширяющей последовательности, равной 80000.

Рис. 4. Границы оценивания задержки ($N_I = 80000$)Fig. 4. Delay Estimation Bounds ($N_I = 80000$)

На рисунке использованы следующие условные обозначения: о – $CRB_{PM}(\hat{\tau})$, * – $CRB_{MSK}(\hat{\tau})$, + – $CRB_\alpha(\hat{\tau})$ для $\alpha = 0,35$.

Граница Крамера – Рао оценки несущей частоты выражается следующим образом [4]:

$$D\{\hat{f}\} \approx \frac{1}{4\pi^2 T_{RMS}^2 q^2}, \quad q \gg 1. \quad (13)$$

Квадрат среднеквадратичной длительности сигнала T_{RMS} в (13) определяется выражением [7]:

$$T_{RMS}^2 = \frac{\int_{-T/2}^{T/2} t^2 |s(t)|^2 dt}{\int_{-T/2}^{T/2} |s(t)|^2 dt}. \quad (14)$$

где $T = KN_I T_S$ – интервал наблюдения (длительность выборки сигнала, накопленного для реализации процедуры оценивания); K – интервал наблюдения, выраженный в количестве информационных символов; $s(t)$ – принимаемый сигнал.

Будем полагать, что интервал наблюдения T много больше длительности интервала информационного символа: $K \gg 1$.

В таком случае по теореме Парсеваля [10] для интеграла в знаменателе выражения (14) можно записать:

$$\int_{-T/2}^{T/2} |s(t)|^2 dt = \int_{-\infty}^{\infty} |G(f)|^2 df.$$

Перепишем числитель выражения (14):

$$\int_{-T/2}^{T/2} t^2 |s(t)|^2 dt = \int_{-T/2}^{T/2} |ts(t)|^2 dt.$$

Если интервал наблюдения $T \gg T_s$, то:

$$\int_{-T/2}^{T/2} |ts(t)|^2 dt \equiv \frac{1}{2\pi} \int_{-\infty}^{\infty} \left| \frac{\partial G(f)}{\partial f} \right|^2 df.$$

Тождество, представленное выше, проистекает из правила дифференцирования преобразования Фурье. В итоге для T_{RMS}^2 можно записать:

$$T_{RMS}^2 = \frac{1}{2\pi} \int_{-\infty}^{\infty} \left| \frac{\partial G(f)}{\partial f} \right|^2 df \bigg/ \int_{-\infty}^{\infty} |G(f)|^2 df. \quad (15)$$

Теперь осталось определить среднеквадратичные длительности для спектров (3, 6 и 7). Выражение (15) с учетом введенного выше ограничения ширины полосы для фазовой модуляции:

$$T_{PM}^2 = T_s^2 \int_{-\pi}^{\pi} \left(\frac{x \cos x - \sin x}{x^2} \right)^2 dx \bigg/ \int_{-\pi}^{\pi} \left(\frac{\sin x}{x} \right)^2 dx.$$

Введем обозначение:

$$T_{MSK}^2 = \frac{2T_s^2}{\pi} \int_{-3/2}^{3/2} \left(\frac{16 \cdot x \cos(\pi x) - 2\pi(1 - 4x^2)\sin(\pi x)}{(1 - 4x^2)^2} \right)^2 dx \bigg/ \int_{-3\pi}^{3\pi} \frac{1 + \cos x}{(1 - (x/\pi)^2)^2} dx. \quad (17)$$

$$\beta_{MSK} = \frac{2}{\pi} \int_{-3/2}^{3/2} \left(\frac{16 \cdot x \cos(\pi x) - 2\pi(1 - 4x^2)\sin(\pi x)}{(1 - 4x^2)^2} \right)^2 dx \bigg/ \int_{-3\pi}^{3\pi} \frac{1 + \cos x}{(1 - (x/\pi)^2)^2} dx. \quad (18)$$

$$D_{MSK}\{\hat{f}\} \approx \frac{1}{4\pi^2 N_I E_S / N_0 \beta_{MSK} T_s^2}. \quad (19)$$

$$CRB_{MSK}(\hat{f}) = D_{MSK}\{\hat{f}\} \cdot T_s^2 \approx \frac{1}{4\pi^2 N_I E_S / N_0 T_s^2 \beta_{PM}}. \quad (20)$$

$$T_{\alpha}^2 = \frac{1}{2\pi} \int_{-T_s(1+\alpha)/2}^{T_s(1+\alpha)/2} \left| \frac{dH(f, \alpha)}{df} \right|^2 df \bigg/ \int_{-T_s(1+\alpha)/2}^{T_s(1+\alpha)/2} |H(f, \alpha)|^2 df. \quad (21)$$

В итоге для соответствующей нормированной границы справедливо:

$$CRB_{\alpha}(\hat{f}) = D_{\alpha}\{\hat{f}\} \cdot T_s^2 \approx \frac{4\alpha}{\pi^3 N_I E_S / N_0}. \quad (22)$$

$$\beta_{PM} = \int_{-\pi}^{\pi} \left(\frac{x \cos x - \sin x}{x^2} \right)^2 dx \bigg/ \int_{-\pi}^{\pi} \left(\frac{\sin x}{x} \right)^2 dx.$$

Граница (13) для фазовой модуляции со спектром (3), ограниченной полосой $\left[-\frac{1}{2T_s}; \frac{1}{2T_s}\right]$, определяется выражением:

$$D_{PM}\{\hat{f}\} \approx \frac{1}{4\pi^2 N_I E_S / N_0 T_s^2 \beta_{PM}}.$$

В итоге нормированная граница Крамера – Рао оценки несущей частоты для фазовой модуляции:

$$CRB_{PM}(\hat{f}) = D_{PM}\{\hat{f}\} \cdot T_s^2 \approx \frac{1}{4\pi^2 N_I E_S / N_0 \beta_{PM}}. \quad (16)$$

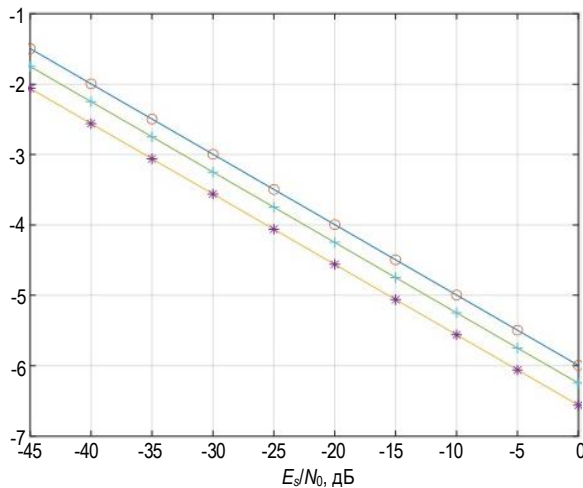
Аналогичным образом можно получить среднеквадратичную длительность сигнала для модуляции минимального сдвига – T_{MSK} . При тех же ограничениях «главного лепестка» T_{MSK}^2 определяет выражение (17). Введем обозначение (18) и получим (19). Отсюда нормированная граница Крамера – Рао оценки несущей частоты для ММС со спектром, ограниченным «главным лепестком», определяется по (20).

Среднеквадратичную длительность сигнала, сформированного фильтром Найквиста – T_{α} можно определить, исходя из выражения (21).

Исходя из (7), можно определить T_{α}^2 и найти соответствующую границу:

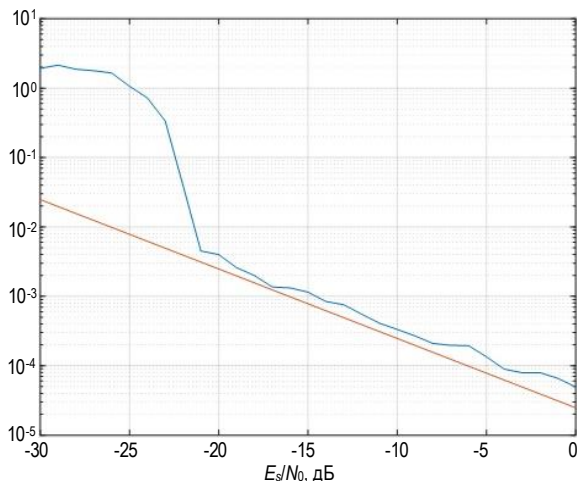
$$D_{\alpha}\{\hat{f}\} \approx \frac{4\alpha}{\pi^3 N_I E_S / N_0 T_s^2}.$$

На рисунке 5 представлены зависимости рассматриваемых границ оценивания несущей частоты (16, 20 и 22) от отношения сигнал/шум на чип для длины расширяющей последовательности, равной 80000.

Рис. 5. Границы оценивания несущей частоты ($N_I = 80000$)Fig. 5. Carrier Frequency Estimation Bounds ($N_I = 80000$)

На рисунке использованы следующие условные обозначения: $\circ$ – $CRB_{PM}(f)$, $*$ – $CRB_{MSK}(f)$, $+$ – $CRB_{\alpha}(f)$ для $\alpha = 0,35$.

В работе [3] был предложен алгоритм оценивания несущей частоты демодулятора сигналов с прямым расширением спектра. Фактически оценивание сводится к вычислению функции правдоподобия принимаемого сигнала на основе быстрого преобразования Фурье и нахождению максимума указанной функции в плоскостях несущей частоты и задержки. Реализацию предложенного алгоритма оценивания для сигнала ММС иллюстрируют зависимости, представленные на рисунке 6.

Рис. 6. ММС: дисперсия оценки несущей частоты ($N_I = 2048$)Fig. 6. MSK: Carrier Frequency Variance ($N_I = 2048$)

Результаты получены для интервала наблюдения K , равного 8 информационным символам. На рисунке также представлена соответствующая граница оценивания. Следует отметить, что полученная дисперсия весьма близка к границе.

Представляется, что алгоритмы оценивания, основанные на нахождении максимума функции неопределенности, фактически реализуют совмест-

ную оценку максимального правдоподобия (МП-оценивание) параметров $\{f, \tau\}$ при условии отсутствия информации о фазе несущей частоты принимаемой информации.

Однако подход к оцениванию, основанный на вычислении функции неопределенности, не реализует оценивание фазы несущего колебания и соответственно не дает определения соответствующей нижней границы. В рассматриваемом случае решение задачи оценивания сводится к оцениванию параметров $\{f, \tau, \phi\}$ в условиях, когда неизвестна передаваемая информация, начало сигнатуры и фаза несущей частоты. Необходимо определить нижние границы дисперсий оценок каждого из элементов вектора $\alpha = \{f, \tau, \phi\}$.

Границы оценивания параметров сигнала, полученные с использованием анализа его дискретных отсчетов на интервале наблюдения, обсуждались в работе [11]. На основе построения информационной матрицы Фишера были получены минимальные граничные дисперсии оценок параметров отрезка гармонического колебания. Предлагаемый подход состоит в следующем: если сигнал характеризуется набором параметров $\alpha = \{\alpha_0, \alpha_1, \alpha_2, \dots\}$, а $\hat{\alpha}_k$ – несмещенная оценка некоторого параметра β_k , то нижняя граница дисперсии оценки α_k определяется элементами матрицы, обратной информационной матрице Фишера [11]:

$$\sigma_k^2 = \text{var}(\hat{\alpha}_k - \alpha_k) \geq J_{kk}^{-1},$$

где J_{kk} – элементы матрицы J^{-1} .

Элементы матрицы Фишера определяются следующим образом [11]:

$$J_{ik} = -E_W \left[\frac{\partial^2 [\ln p(\tilde{s}_k | \alpha)]}{\partial \alpha_i \partial \alpha_k} \right],$$

где E_W – статистическое усреднение по отношению к шуму; $p(\tilde{s}_k | \alpha)$ – совместная функция плотности вероятности вектора $\tilde{s}_k$ для заданных параметров $\alpha = \{\alpha_0, \alpha_1, \alpha_2, \dots\}$; $\ln p(\tilde{s}_k | \alpha)$ – логарифм функции правдоподобия.

В работе [12] предложенный подход к определению потенциальных границ оценивания параметров сигналов развит для сигналов с различными видами модуляции. Показано, что компоненты матрицы Фишера могут быть представлены в виде:

$$\mathbf{F} = E_d \left\{ \begin{pmatrix} J_{ff} & J_{f\phi} & J_{f\tau} \\ J_{\phi f} & J_{\phi\phi} & J_{\phi\tau} \\ J_{\tau f} & J_{\tau\phi} & J_{\tau\tau} \end{pmatrix} \right\}, \quad (23)$$

где $E_d\{\cdot\}$ – усреднение элементов матрицы по информационной последовательности.

Заметим, что в матрице (23) номера i, k заменены на индексы, соответствующие параметрам $\{f, \tau, \phi\}$. В итоге в [12] показано, что матрица Фишера для параметров $\alpha = \{f, \tau, \phi\}$ принимает следующий вид:

$$\mathbf{F} = \begin{pmatrix} F_{ff} & F_{f\varphi} & 0 \\ F_{\varphi f} & F_{\varphi\varphi} & 0 \\ 0 & 0 & F_{\tau\tau} \end{pmatrix}.$$

Отсюда нижняя граница дисперсии оценки несущей частоты определяется следующим образом:

$$\text{var}(\hat{f} - f) = \frac{1}{F_{ff} - F_{f\varphi}^2 / F_{\varphi\varphi}}. \quad (24)$$

Будем полагать, что сигнал задан на интервале наблюдения в виде отсчетов с периодом следования T_d . С учетом введенного выше ограничения полосы сигнала $T_d < T_s/2$. Пусть N – интервал наблюдения, выраженный в количестве интервалов T_d . Будем полагать, что интервал наблюдения содержит целое количество информационных символов. То есть $N = KN_I$. В таком случае для (24) можно записать [12]:

$$\text{var}(\hat{f} - f) = \frac{3}{2\pi^2 E_s / N_0 T_s^2 (KN_I - 1)^2 KN_I} \times \frac{1}{1 + O(N^{-2})},$$

где $O(N^{-2})$ – величина порядка N^{-2} .

Понятно, что при $K \gg 1$ для нормированной границы оценки несущей частоты справедливо:

$$\text{CRLB}(\hat{f}) = \text{var}(\hat{f} - f) T_s^2 = \frac{3}{2\pi^2 E_s / N_0 N_I^3 K^3}. \quad (25)$$

Выражение (25) определяет нижнюю границу Крамера – Рао оценки несущей частоты (CRLB, аббр. от Cramer – Rao Lower Bound). В некоторых работах эту границу называют модифицированной нижней границей Крамера – Рао [9, 13] (MCRLB, аббр. от Modified Cramer-Rao Lower Bound).

Для иллюстрации соотношения границ (16 и 25) на рисунке 7 представлены зависимости $\text{CRB}_{PM}(\hat{f})$ и $\text{CRLB}(\hat{f})$ от отношения сигнал/шум на чип E_s/N_0 , где $\circ$ – $\text{CRB}_{PM}(f)$, * – $\text{CRLB}(f)$.

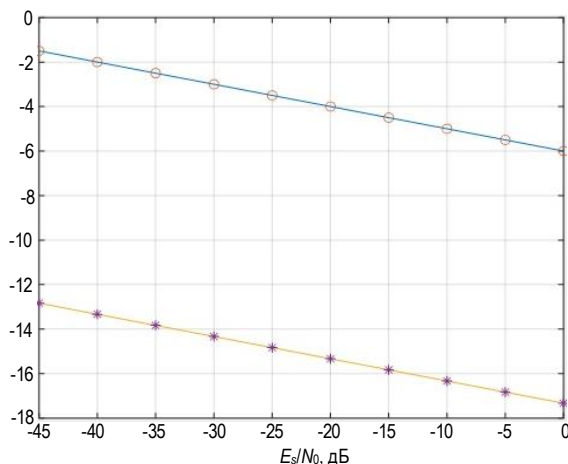


Рис. 7. Границы оценивания несущей частоты ($N_I = 80000$, $K = 4$)

Fig. 7. Carrier Frequency Estimation Bounds ($N_I = 80000$, $K = 4$)

Естественно, граница (16) существенно уступает нижней границе Крамера – Рао. Это объясняется тем, что при получении МП-оценки несущей частоты на основе вычисления функции правдоподобия не использовалась информация о фазе несущего колебания, а нижняя граница Крамера – Рао – это граница, полученная при предположении, что все параметры, кроме несущей частоты, известны.

Нижняя граница оценки задержки определяется следующим образом [12]:

$$\text{var}(\hat{\tau} - \tau) = \frac{1}{F_{\tau\tau}} = \frac{1}{2E_s / N_0 KN_I W_{RMS}^2}.$$

Так, например, для сигнала с фазовой модуляцией с учетом (4), можно записать:

$$\text{CRLB}_{PM}(\hat{\tau}) = \frac{2\pi}{E_s / N_0 KN_I} \int_{-\pi}^{\pi} \left(\frac{\sin x}{x} \right)^2 dx.$$

Аналогично для ММС:

$$\text{CRLB}_{MSK}(\hat{\tau}) = \frac{8\pi^2}{\xi_{MSK} E_s / N_0 KN_I}$$

и для сигналов с «найквистовским» спектром:

$$\text{CRLB}_\alpha(\hat{\tau}) = \frac{1}{2E_s / N_0 KN_I \left(\frac{1}{12} + \alpha^2 \left(\frac{1}{4} - \frac{2}{\pi^2} \right) \right)}.$$

Заметим, что при получении границ оценивания для сигналов с «найквистовским» спектром не ставится ограничение на виды модуляции принимаемого сигнала. Т. е. соответствующие границы справедливы для сигналов различных видов модуляции, сформированных с использованием спектра Найквиста.

И, наконец, нижняя граница оценки фазы несущего колебания определяется как [12]:

$$\text{CRLB}(\hat{\varphi}) = \text{var}(\hat{\varphi} - \varphi) = \frac{1}{2E_s / N_0 KN_I}.$$

Следует отметить, что оценивание фазы на практике реализуется в процессе демодуляции сигнала. Поэтому практическую значимость имеет связь между нижней границей оценки фазы несущей и полосой соответствующей петли синхронизации демодулятора. Интервал наблюдения связан с полосой указанной петли следующим образом [9]:

$$N = KN_I = \frac{1}{2B_\varphi T_b},$$

где B_φ – односторонняя полоса петли синхронизации демодулятора по фазе несущего колебания.

В результате получим выражение, связывающее полосу петли и нижнюю границу Крамера – Рао оценки фазы:

$$\text{CRLB}(\hat{\varphi}) = \frac{B_\varphi T_b}{E_s / N_0}.$$

Выводы

Определены потенциальные границы дисперсии оценок параметров сигнала с прямым расширением спектра. Представлены аналитические выражения для границ Крамера – Рао оценки несущей частоты и задержки для сигналов фазовой модуляции, модуляции минимального сдвига и сигналов, сформированных с использованием спектра Найквиста. Получены выражения для нижней границы Крамера – Рао, определяющие минимальные границы дисперсий оценок несущей частоты, задержки и фазы несущего колебания. В частности, получено выражение, связывающее нижнюю границу Крамера – Рао оценки фазы несущего колебания, как зависимость соответствующей граничной дисперсии и нормированной полосы петли $B_{\phi}T_b$.

Практическая значимость полученных результатов состоит в представленных зависимостях границ оценивания параметров сигналов с прямым расширением спектра от отношения сигнал/шум на чип длины расширяющей последовательности и длительности интервала наблюдения.

Результаты позволяют оценить эффективность получаемых оценок для широкого круга задач, связанных с реализацией алгоритмов синхронизации демодуляторов сигналов с прямым расширением спектра.

Предложенный подход позволяет получать границы оценивания параметров широкого круга сигналов с различными видами модуляции и спектральными характеристиками.

Естественно, что при реализации алгоритмов оценивания параметров сигналов с прямым расширением спектра необходимо сопоставлять получаемые дисперсии оценок с границами, определенными в представленной работе. В частности, в

настоящей работе показано, что граница (20) практически достигается при реализации совместного МП-оценивания параметров $\{f, \tau\}$ сигнала с прямым расширением спектра, использующего модуляцию минимального сдвига. При этом полученные дисперсии оценок несущей частоты существенно уступают соответствующей нижней границе Крамера – Рао.

Соответствующий алгоритм оценивания предложен в работах [2, 3], но оценка задержки в работах практически не обсуждается. В перспективе представляется целесообразным разработка и исследования алгоритмов эффективного оценивания как несущей частоты, так и задержки. Что касается оценивания фазы несущего колебания, основной акцент следует сделать на разработке собственно алгоритмов демодуляции сигнала.

Естественно, что во всех случаях интерес вызывают алгоритмы оценивания, обеспечивающие дисперсии, близкие к соответствующим нижним границам Крамера – Рао, представленным в настоящей работе.

Подводя итог проведенным исследованиям, можно сделать следующие выводы:

- получены аналитические выражения для границ оценивания параметров сигналов с прямым расширением спектра;
- предложен подход, позволяющий получить границы оценивания параметров сигналов для широкого круга задач;
- полученные результаты позволяют определить границы оценивания в зависимости от видов модуляции, спектральных характеристик принимаемых сигналов, длины расширяющей последовательности и длительности интервалов наблюдения;
- в качестве иллюстрации результатов в работе представлены графики, показывающие соотношения обсуждаемых границ оценивания.

Список источников

1. Брусин Е.А. Начальная синхронизация демодулятора сигнала с прямым расширением спектра с использованием быстрого преобразования Фурье // 11-я Международная научно-техническая и научно-методическая конференция «Актуальные проблемы инфотелекоммуникаций в науке и образовании» (АПИНО-2022, Санкт-Петербург, Россия, 15–16 февраля 2022). СПб.: СПбГУТ, 2022. Т. 2. С. 522–527.
2. Брусин Е.А. Реализация алгоритма начальной синхронизации демодулятора сигналов с прямым расширением спектра на основе быстрого преобразования Фурье. Часть 1. Постановка задачи и подход к решению // Труды учебных заведений связи. 2022. Т. 8. № 4. С. 21–27. DOI:10.31854/1813-324X-2022-8-4-21-27
3. Брусин Е.А. Реализация алгоритма начальной синхронизации демодулятора сигналов с прямым расширением спектра на основе быстрого преобразования Фурье. Часть 2. Оценивание несущей частоты // Труды учебных заведений связи. 2023. Т. 9. № 1. С. 35–40. DOI:10.31854/1813-324X-2023-9-4-35-40
4. Ipatov V.P. Spread Spectrum and CDMA. Principles and Applications. John Wiley & Sons, 2005.
5. Бердников А.С., Жуков Е.Т., Носов Е.В. Проект модема сличения шкал времени по дуплексному каналу спутниковой связи // Труды института прикладной астрономии РАН. 2016. № 36. С. 21–26.
6. Интерфейсный контрольный документ. Радиосигналы и состав цифровой информации функционального дополнения системы ГЛОНАСС системы дифференциальной коррекции и мониторинга (редакция 1) // Система дифференциальной коррекции и мониторинга. URL: <https://sdcm.ru> (дата обращения 22.11.2023)
7. Cariolaro G. Unified Signal Theory. Springer Science & Business Media, 2011. 928 p.
8. Банкет В.М., Дорофеев В.М. Цифровые методы в спутниковой связи. М.: Радио и Связь, 1988. 240 с.
9. Mengali U. Synchronization Technique for Digital Receivers. Springer Science & Business Media, 1997. 520 p.

10. Сиберт У.М. Цепи, сигналы, системы: В 2-х частях. М.: Мир, 1988. 696 с.
11. Rife D.C., Boorstyn R.R. Single tone parameter estimation from discrete-time observation // IEEE Transactions on Information Theory. 1974. Vol. 20. Iss. 5. PP. 591–598. DOI:10.1109/TIT.1974.1055282
12. Meyer H., Moeneclaye M., Fechtel S.A. Digital Communication Receivers. John Wiley&Sons, Inc., New York, 1998.
13. Gini F., Reggiannini R., Mengali U. The modified Cramer-Rao bound in vector parameter estimation // IEEE Transactions on Communications. 1998. Vol. 46. PP. 52–60. DOI:10.1109/26.655403

References

1. Brusin E.A. Initial Timing of a Direct Spread Demodulator Using Fast Fourier Transform. *Proceedings of the XIth International Conference on Infotelecommunications in Science and Education, 15–16 February 2022, St. Petersburg, Russia, vol.2*. St. Petersburg: The Bonch-Bruевич Saint-Petersburg State University of Telecommunications Publ.; 2022. p.522–527.
2. Brusin E. Direct Sequence Spread Spectrum Signal's Demodulator Acquisition Implementation Based on Fast Fourier Transform. Part 2. Problem Statement and Solution Approach. *Proceedings of Telecommun. Univ.* 2022;8(4):21–27. DOI:10.31854/1813-324X-2022-8-4-21-27
3. Brusin E. Direct Sequence Spread Spectrum Signal's Demodulator Acquisition Implementation Based on Fast Fourier Transform. Part 3. Carrier Frequency Estimation. *Proceedings of Telecommun. Univ.* 2023;9(1):35–40. DOI:10.31854/1813-324X-2023-9-1-35-40
4. Ipatov V.P. *Spread Spectrum and CDMA. Principles and Applications*. John Wiley & Sons; 2005.
5. Berdnikov A.S., Zhukov E.T., Nosov E.V. Project of the modem of time scale comparison on duplex channel of satellite communication. *Transactions of the Institute of Applied Astronomy RAS*. 2016;36:21–26.
6. *System of Differential Correction and Monitoring*. Interface control document. Radio signals and composition of digital information of the functional completion of the GLONASS system of differential correction and monitoring (revision 1). URL: <https://sdcm.ru> [Accessed 22.11.2023]
7. Cariolaro G. *Unified Signal Theory*. Springer Science & Business Media; 2011. 928 p.
8. Banquet V.M., Dorofeev V.M. *Digital methods in satellite communications*. Moscow: Radio i svyaz Publ.; 1988. 240 p.
9. Mengali U. *Synchronization Technique for Digital Receivers*. Springer Science & Business Media; 1997. 520 p.
10. Sibert W.M. *Circuits, Signals, Systems*: In 2 parts. Moscow: Mir Publ.; 1988. 696 p.
11. Rife D.C., Boorstyn R.R. Single tone parameter estimation from discrete-time observation. *IEEE Transactions on Information Theory*. 1974;20(5):591–598. DOI:10.1109/TIT.1974.1055282
12. Meyer H., Moeneclaye M., Fechtel S.A. *Digital Communication Receivers*. John Wiley&Sons, Inc., New York; 1998.
13. Gini F., Reggiannini R., Mengali U. The modified Cramer-Rao bound in vector parameter estimation. *IEEE Transactions on Communications*. 1998;46:52–60. DOI:10.1109/26.655403

Статья поступила в редакцию 09.11.2023; одобрена после рецензирования 01.12.2023; принята к публикации 04.12.2023.

The article was submitted 09.11.2023; approved after reviewing 01.12.2023; accepted for publication 04.12.2023.

Информация об авторе:

БРУСИН
Ефим Александрович

кандидат технических наук, руководитель проектного направления АО «Российский институт радионавигации и времени», доцент кафедры электроники и схемотехники Санкт-Петербургского государственного университета телекоммуникаций им. проф. М.А. Бонч-Бруевича

 <https://orcid.org/0000-0002-6742-2705>

Научная статья

УДК 621.396

DOI:10.31854/1813-324X-2023-9-6-34-41



Electromagnetic Accessibility of Starlink User Terminals

Sergei V. Dvornikov, practicdsv@yandex.ru

Saint Petersburg State University of Aerospace Instrumentation,
St. Petersburg, 190000, Russian Federation
Military Academy of Communications,
St. Petersburg, 194064, Russian Federation

Abstract: The results of assessing the electromagnetic accessibility of radio emission sources of the Starlink satellite communication system are presented. The features of the organizational and technical structure of the system are considered. Its main technical capabilities for organizing the provision of telecommunications services to users of subscriber terminals are analyzed. The technical features of the infrastructure of the Starlink satellite communication system, which are essential for the electromagnetic accessibility of its sources, have been studied. The possibility of radio links for transmitting information content in the upstream and downstream channels of the Starlink satellite communication system was assessed. The main stages of the developed methodology for assessing the electromagnetic accessibility of sources are presented. Analytical expressions are given for calculating the probability of detection and attenuation of signals. The requirements for the sensitivity of the receiving equipment of monitoring equipment are substantiated. Dependences of the level of signal attenuation on the distance of control equipment were obtained.

Keywords: assessment of electromagnetic accessibility of radio emission sources, Starlink satellite communication system, probability of correct signal detection, level of signal attenuation, sensitivity of receiving devices

For citation: Dvornikov S.V. Electromagnetic Accessibility of Starlink User Terminals. *Proceedings of Telecommun. Univ.* 2023;9(6):34–41. DOI:10.31854/1813-324X-2023-9-6-34-41

Электромагнитная доступность абонентских терминалов Starlink

Сергей Викторович Дворников, practicdsv@yandex.ru

Санкт-Петербургский государственный университет аэрокосмического приборостроения,
Санкт-Петербург, 190000, Российская Федерация
Военная академия связи им. С.М. Буденного,
Санкт-Петербург, 194064, Российская Федерация

Аннотация: Представлены результаты оценки электромагнитной доступности источников радиоизлучений спутниковой системы связи Starlink. Рассмотрены особенности организационно-технического построения системы. Проанализированы основные ее технические возможности по организации предоставления телекоммуникационных услуг пользователям абонентских терминалов. Исследованы технические особенности инфраструктуры спутниковой системы связи Starlink, существенные для электромагнитной доступности ее источников. Оценена возможность радиолиний по передаче информационного контента в восходящих и нисходящих каналах спутниковой системы связи Starlink. Представлены основные этапы разработанной методики оценки электромагнитной доступности источников. Приведены аналитические выражения для расчета вероятности обнаружения и затухания сигналов. Обоснованы требования к чувствительности приемной аппаратуры средств контроля. Получены зависимости уровня затухания сигнала от удаленности средств контроля.

Ключевые слова: оценка электромагнитной доступности источников радиоизлучений, спутниковая система связи Starlink, вероятность правильного обнаружения сигнала, уровень затухания сигнала, чувствительность приемных устройств

Ссылка для цитирования: Дворников С.В. Электромагнитная доступность абонентских терминалов Starlink // Труды учебных заведений связи. 2023. Т. 9. № 6. С. 34–41. DOI:10.31854/1813-324X-2023-9-6-34-41

General information about the Starlink Satellite Communications System

The development of the project began in 2015, and already in February 2018, the launch of two experimental spacecraft (SC) Tintin-A and Tintin-B into low-Earth orbit, located at an altitude of about 1150 km, was successfully completed. Initially, their task was to test the developed data transmission technologies in the Ku and Ka-bands under the Starlink program [1].

By April 2019, based on the results of the test tests, the need to change the initially planned orbits was justified by reducing their altitude to 550 km and organizing spacecraft flights in 24 orbital planes with an inclination of 53° (<https://www.eoportal.org/satellite-missions/starlink>, <https://habr.com/ru/post/652467>).

The first test version of the full-scale operation of SSC Starlink, the grouping of which included 60 SC Starlink version 1.0, was tested by November 2019. And in October 2020, SpaceX announced its readiness to provide communication services throughout the Earth (<https://www.comnews.ru/content/209438/2020-10-07/2020-w41/enciklopediya-starlink>). At the same time, the emphasis was on providing high-speed access (up to 500 Mbit/s, in the absence of other subscribers in this cell) to the Internet up to an unlimited amount of traffic with a delay of no more than 20-40 ms. In August 2022, SpaceX announced a project developed jointly with mobile communications company T-Mobile to provide mobile communications services based on SSC Starlink. As a result, by the beginning of September 2022, the number of SSC Starlink users exceeded 500,000 subscriber numbers (<https://en.wikipedia.org/wiki/Starlink>). As the main service, the Starlink Business communication package was provided at Internet access speeds from 150 to 500 Mbit/s (<https://naked-science.ru/article/tech/starlink-perehvat>), [2]. As of December 2022, the number of users exceeded more than 1 million, and by May 2023 it reached 1.5 million subscribers and reached 2 million subscribers by September 2023.

Initially, it was planned that the implementation of the project would provide subscribers with telephone communications in areas not covered by terrestrial cellular networks, for which an additional set of PCS (personal communications service) equipment, presumably of the IMT-2000 standard, was installed on the Starlink spacecraft. This approach made it possible, at a speed of 2 to 4 Mbit/s, to organize the simultaneous operation of about 2000 voice calls and more than 1 million short

text messages in the SSC Starlink coverage area. However, later it was decided that it was necessary to work together with standard cell phones, for which IP traffic was organized.

Currently, SSC Starlink communication services are provided in 70 countries (including Ukraine) (https://en.wikipedia.org/wiki/Starlink_in_the_Russo-Ukrainian_War). Between February 2018 and February 2023, SpaceX successfully launched 4,002 Starlink satellites into orbit, including prototypes and satellites that later failed or deorbited before entering service (<https://www.eoportal.org/satellite-missions/starlink>). Spacecraft are placed in orbits at an altitude of 540-570 km. Data on the general development prospects of SSC Starlink are presented in Table 1 (<https://naked-science.ru/article/tech/starlink-perehvat>).

TABLE 1. General Planned Composition of SC SSC Starlink Orbital Constellation of the 1st Phase

Generation of satellites	Orbital titude, km	Orbital inclination, degrees	Number of orbital planes	Number of SC in plane	Total SC in this group
Group 1	550	53	72	22	1584
Group 2	570	70	36	20	720
Group 3	560	97,6	6	58	348
Group 4	540	53,2	72	22	1584
Group 5	560	97,6	4	43	172
Total in the 1st phase of the orbital constellation					4408

And if half of the entire satellite constellation was deployed at the beginning of March 2022, then the full deployment is planned to be completed by the end of March 2027.

Further development of SSC Starlink involves a gradual transition to the second phase of deployment of the orbital constellation using SC Starlink versions 1.5 and 2.0. The main payload of SC Starlink is 2 antenna complexes for communication with gateway stations (GS) (gateways) and user terminals (UT).

The antenna complex for communication with the GS (or feeder line) is a parabolic antenna, aimed during the flight at the point on Earth where the GS is located. The system operates in the Ka-band (18/30 GHz). The characteristics of SSC Starlink communication channels are presented in Table 2 [1]. Each SC has at its disposal a frequency resource of 2100 MHz allocated in the direction from the gateway station to the SC, and a frequency resource of 1300 MHz allocated in the reverse direction. When used on left- and right-polarized lines (in

the case of using circularly polarized antennas), it becomes possible to transmit traffic in a common band of 4200 MHz from GS to SC and 2600 MHz in the opposite direction.

Communication between UT and SC is carried out in the Ku-band (<https://en.wikipedia.org/wiki/Starlink>). In this case, a frequency band of 2000 MHz can be used to transmit information from SC to UT, and only 500 MHz from UT to SC. Taking into account the possibility of using two polarizations, the frequency range for transmitting SC traffic will be 4000 MHz for downstream transmission, and 1000 MHz for reception.

It should be noted that on board the SC there is a set of equipment for organizing the operation of the commandradio link and telemetry transmission, using a 150 MHz band in the Ka and Ku bands, respectively.

TABLE 2. Characteristics of SSC Starlink Communication Channels

Type of communication channel and direction of reception and transmission	Frequency ranges, GHz	Available MHz in one
Service: downlink (SC – UT)	10.7–12.7	2000
Service: downlink (SC – GS)	17.8–18.6 18.8–19.3	800 500
Service: uplink (UT – SC)	14.0–14.5	500
Service: uplink (GS – SC)	27.5–29.1 29.5–30.0	1600 500
Telemetry and Down Control (downlink SC – control station)	12.15–12.25 18.55–18.60	100 50
Telemetry and upstream control (uplink control station – SC)	13.85–14.00	150

All Starlink SCs operate in repeater mode, i.e. without processing information. On board, only the frequency of the received signal is changed (converted) and amplified (<https://naked-science.ru/article/tech/starlink-perehvat>).

In addition, the 1st generation SC does not provide for communication between satellites (ISL – Inter Satellite Link), so they can only receive and transmit information to the GS located on Earth.

4 earth stations are declared as TT&C stations (control, monitoring, telemetry reception), including the Brustner teleport, located in Washington state. The Starlink spacecraft is within the visibility range of the TT&S station for no more than five minutes, while the volume of data collected from the SC orbital constellation as of June 2020 reached 5 TB per day, i.e. at least 10 GB from one SC per day (<https://en.wikipedia.org/wiki/Starlink>). On board each SC Starlink there are about 70 separate processors running the Linux operating system and about 10 microcontrollers.

Being in low-Earth orbit, at an altitude of 550 km, the SC is capable of "covering" with its signal a spot on the Earth's surface with a radius of about 950 km (i.e., a diameter of about 1900 km), provided that the elevation angle for the UT is at least 25° [1].

Separately, it should be noted that effective operation of antennas with a flat phased array is possible only at an elevation angle of 40° or more.

The initial development of SSC Starlink was focused on the use of directional antennas, the characteristics of the radiation patterns (RP) of which are given in Table 3 [1]. Table 3 correspond to the conditions according to which the diameter of the beam spot on the earth's surface is 45 km, which corresponds to the angle RP of the SC beam of 4.5° (when deviating from the nadir line, the angle can vary from 3 to 5 degrees, the further from the nadir line, the greater corner); EIRP – equivalent isotropic radiated power.

Theoretically, each Starlink SSC is capable of forming up to 300 access zones for UTs, taking into account left and right polarization. In addition, it can operate in two frequency bands, provided that the maximum tilt angle of the UT antenna system is not less than 25°.

TABLE 3. Characteristics of Up and Down Lines

Channel type	Frequency	Modulation type	Maximum EIRP	Half Power RP Beam Width
Downlink (SC – UT)	10.7–12.7 GHz	from QAM-2 to QAM-64	No data	3.5° (boresight) 5.5° (at slant)
Uplink (UT – SC)	14.0–14.5 GHz	from QAM-2 to QAM-64	38.2 dBW	2.8° (boresight) 4.5° (at slant)

In Fig. 1 shows the topology of SC Starlink availability zones for UTs declared by SpaceX [1].

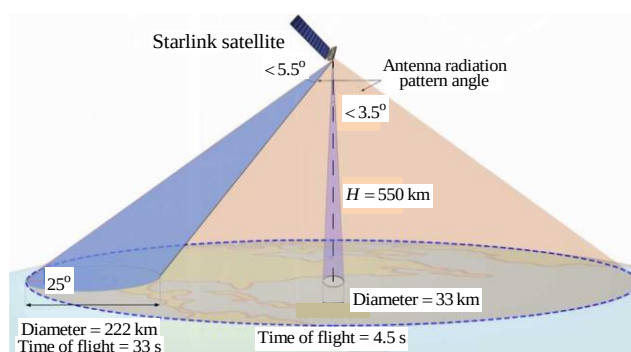


Fig. 1. Starlink SSC Availability Area Topology for UT, Formed by RP SC

The generated downlink channel from SC to UT has a maximum width of 240 MHz, while the uplink channel bandwidth is only 60 MHz (https://pikabu.ru/story/yentsiklopediya_starlink_ot_spacex_rozhdenie_struktura_i_razvitiye_ch2_7762860). This configuration is optimal from the point of view of efficient use of the frequency resource, since as a result, no more than 16 beams are formed within the coverage area of one SC, which fully use the available frequency resource of the Ku-band, equal to 4000 MHz. These values are obtained taking into account the guard intervals and frequencies used for the radio command link and telemetry transmission, in the case of using both types of polarizations

when transmitting information content from SC to UT, see Fig. 2 [1]. According to the configuration the Starlink SC SSC accessibility zone with an elevation angle of up to 25° at an orbital altitude of 550 km is a circle with a diameter of approximately 1900 km. Accordingly, the area of such a zone is exactly 2,835,294 sq. km. Then, with an orbit altitude of 550 km, the SC flight time of the UT accessibility zone is only 4.1 minutes or about 250 s [2].

An analysis of the topology of the UT availability zone shows that the greatest efficiency of the SC – UT radio link is ensured when subscribers are not at the nadir (sub-satellite point), but in the peripheral visibility zone, even taking into account the fact that there the effective antenna area is reduced, which in turn leads to a decrease in radio link capacity. These features are due to the possibility of rotating the UT antenna towards the SC to ensure an optimal angle between the phased array plane and the direction towards the SC (ideally 90°).

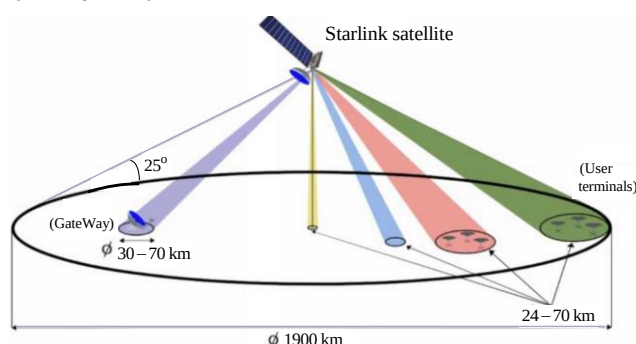


Fig. 2. SC SSC Starlink Access Configuration to UT

The overall coordination and management of the entire SSC Starlink, GS and UT network is carried out by the Network Control Center (NCC). Due to the lack of information about the NCC, it can be assumed that its basis is a set of servers interconnected by fiber optic communication lines with GS. The NCC complex also in-

cludes the ground-based complex for the Control and Collection of Telemetry of the Starlink network, consisting of 4 teleports with installed stations for the control and collection of telemetry systems in the Ku and Ka-band [1]. Geographically, they are located in Brewster (Washington State in the USA), Cordova (Argentina), Tromso (Norway), Avarua (New Zealand).

The telemetry and control channel for each SC can be active for up to 2.5 hours per day (12 minutes per revolution around the Earth) with an estimated telemetry session time of 60 minutes per day (or 10 minutes per orbit) (https://pikabu.ru/story/yentsiklopediya_starlink_ot_spacex_rozhdenie_struktura_i_razvitiye_ch2_7762860).

The basis for UT access to the Internet is GS, which ensures the transmission of information through SC. In the absence of communication with the SC, for the UT to function, it is necessary that at least one GS be located in the SC access area, capable of simultaneously working with hundreds and thousands of UTs. A typical GS has 8 antennas, each of which is capable of transmitting information to its "own" SC.

Currently, SSC Starlink on GS uses 50 W transmitters and parabolic antennas with a mirror diameter of 1.5 m. These antennas, unlike phased arrays, can operate at low elevation angles (according to SpaceX – up to 5°). At the same time, the terminal provides a channel capacity of 500 MHz (including guard intervals, up to 480 MHz). And the equivalent isotropic radiated power (EIRP) reaches 66.5 dBW (<https://en.wikipedia.org/wiki/Starlink>).

The operational capabilities of UT SSC Starlink are much more modest. Initially, SpaceX announced 5 types of UT, which are positioned as Model "ES, A, (B, C, D, E)". Technical characteristics of UT are presented in Table 4 (<https://www.comnews.ru/content/209438/2020-10-07/2020-w41/enciklopediya-starlink>). The first 2 columns refer to earth stations for satellite control and monitoring tasks, and the last five are US.

TABLE 4. Technical Characteristics of SSC Starlink Terminals

Elements	Model "Space* Telem Ku/Ka"	Model "Space* Telem X"	Model "ES A"	Model "ES – B"	Model "ES – C"	Model "ES – D"	Model "ES – E"
Antenna name and model number	CGC Technology T450	Orbital Systems 3.7 Meter	SpaceX	SpaceX	SpaceX	SpaceX	CGC Technology T450
Antenna type	Cassegrain parabolic antenna	Parabolic antenna with main focus	Phased array antenna	Phased array antenna	Type unknown	Parabolic antenna	Cassegrain parabolic antenna
Frequency range	Operation over the entire range of declared frequencies	2.0–2.1 GHz uplink 7.2–8.4 GHz descending channel	Operation over the entire range of declared frequencies	Operation over the entire range of declared frequencies	Operation over the entire range of declared frequencies	Operation over the entire range of declared frequencies	Operation over the entire range of declared frequencies
Maximum Gain / RP width at 3 dB roll-off level	Uplink (Ku): 26 dB 0.22 ° Descending line (Ku): 22 dB 0.3 ° Downward channel (Ka): 27 dB 0.2 °	Uplink: 8.9 dB; 1.7. Downward channel: 16.8 dB; 0.8 °	3 dB at 3.5 ° full beam	7 dB at 2 ° full beam	6 dB at 2.4 ° full beam	16 dB at 0.9 ° full beam	Uplink: 26 dB; 0.22 ° Downward channel: 22 dB; 0.3 °

In Table 4 Model "Space* Telem Ku/Ka" and Model "Space* Telem X" belong to GS (used for management and monitoring of SC). UTs on the Starlink network can operate receive channels (downlink) in the frequency range 15, 30, 60, 120 and 240 MHz. This distribution allows for symbol rates of 15, 30, 60, 120 and 240 MBd/s (https://pikabu.ru/story/yentsiklopediya_starlink_ot_spacex_rozhdenie_struktura_i_razvitiye_ch2_7762860). At the same time, the information speed of the incoming stream reaches 350 Mbit/s, and the outgoing one – 130 Mbit/s. Typically, UT operation is carried out in the 60 MHz band. And with an effective antenna diameter of 48 cm and an RP angle of 2.8°, an EIRP of 38.2 dBW is provided, see Table 4. The power of the UT transmitter can vary depending on its inclination relative to the zenith line. In the case when the antenna beam is directed to the zenith, the power supplied to the antenna is 0.76 W (with a maximum deviation from the vertical of 4.06 W).

The spectral efficiency of the UT for reception with a channel width of 240 MHz and a transmission rate of 350 Mbit/s is only 1.5 bits/Hertz.

Methodology for Assessing EMA of SSC Starlink Subscribers

When assessing the Electromagnetic Availability of terminals of the Starlink Satellite Communication System, we will determine as the main indicator the probability of detecting the P_{det} signal in the receiving paths of monitoring equipment.

Next, we introduce the following assumptions:

1) the signal detection process (distribution of amplitude values of detected signals) obeys the normal distribution law:

$$\Delta F_{res} T_{av} \gg 1, (\Delta F_{res} T_{av} \geq 10), \quad (1)$$

where ΔF_{res} – receiver path bandwidth, Hz; T_{av} – process averaging time, s;

2) at the first stage, only the internal noise of the receiver is taken into account;

3) signal spectrum width ΔF_s , Hz, coordinated with the bandwidth of the control means receiving path ΔF_{res} , i.e. $\Delta F_s = \Delta F_{res}$.

Under the assumptions made, the value of P_{det} can be calculated using the following formula [3–5]:

$$P_{det} = \Phi \left(\frac{h^2 \sqrt{\Delta F_{res} T_{av}} - \Phi^{-1}(1 - P_{fa})}{1 + h^2} \right), \quad (2)$$

where h^2 – signal-to-noise ratio in terms of power at the receiver input, dB; P_{fa} – false detection probability; $\Phi(x) = 1/\sqrt{2\pi} \int_{-\infty}^x \exp(-t^2/2) dt$ – probability integral; $\Phi^{-1}(x)$ – inverse function $\Phi(x)$.

To solve the detection problem, we determine the value of P_{fa} in the range of values from 10^{-6} to 10^{-3} .

As initial data we will assume:

- signal bandwidth equal $\Delta F_s = 60$ MHz;
- duration of signal processing no more 33 mks;

- process average value $T_{av} = 40$ mks;
- probability of false alarm $P_{fa} = 10^{-3}$.

Taking into account the accepted initial data and the assumptions made, expression (2) can be transformed to the following form:

$$P_{det} = \Phi \left(\frac{4,9 h^2 - 3,2}{1 + h^2} \right), \quad (3)$$

where $h^2 = P_s/\sigma_N^2$ – ratio of signal power P_s to power/noise spectral density σ_N^2 (SNR) at the control receiver input.

In Fig. 3 shows a graph of the probability of detecting the SSC Starlink signal in the 60 MHz band according to the Neyman-Pearson criterion at the false alarm probability value $P_{fa} = 10^{-3}$ [6].

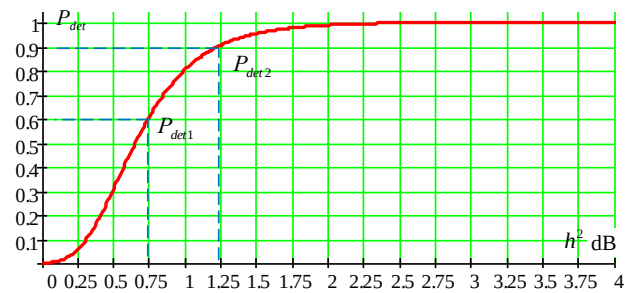


Fig. 3. Dependence of the Probability of Detecting a Starlink SSC Signal on the SNR Value in the Channel

In Fig. 3 shows two points corresponding to the probability values $P_{det1} = 0.6$ and $P_{det2} = 0.9$.

Note that for the conditions under consideration the probability $P_{det1} = 0.6$ is ensured at an SNR value of the order $h_1^2 = -1.25$ dB. And the probability $P_{det2} = 0.9$ provided at SNR value $h_2^2 = 1$ dB.

The obtained result allows us to conclude that, theoretically, SSC Starlink signals are quite easily detected, despite the wide operating frequency range.

Next, we will evaluate the conditions for radio monitoring of UT signals during operation of the uplink UT – SC. Thus, the power of the SSC Starlink signal at the receiving point can be estimated by the following expression [7, 8]:

$$P_s = \frac{P_{tr} G_{tr}(f_s) G_a(f_s) \lambda^2}{(4\pi R)^2}, W, \quad (4)$$

or else

$$P_s = P_{tr} + G_{tr}(f_s) + G_a(f_s) + 20 \lg(\lambda^2) - 20 \lg(R) - 22, \text{ dB}, \quad (5)$$

where P_{tr} – SSC Starlink terminal transmitter power, dB; $G_{tr}(f_s)$, $G_a(f_s)$ – antenna gains of the SSC Starlink terminal and radio control receiver, dB; λ – signal emission wavelength, m; R – distance from the SSC Starlink terminal to the location of radio control facilities, m.

Next, let's estimate the noise level:

$$\sigma_N^2(\Delta F_{res}) = \sigma_{n.a.}^2(\Delta F_s) + \sigma_{n.res.}^2(\Delta F_s), W/Hz, \quad (6)$$

where $\sigma_N^2(\Delta F_{res})$ – noise power dispersion; $\sigma_{n.a}^2(\Delta F_s)$ – antenna noise dispersion in band ΔF_s , W/Hz; $\sigma_{n.res}^2(\Delta F_s) = N_{n.res}^2(f_s)\Delta F_s$ – noise dispersion in the signal reception band ΔF_s , W/Hz; $N_{o.n}^2(f_s)$ – noise power spectral density at frequency f_s , dBm (dB relatively 1 mW/Hz) in other ways dB (dB relatively 1 W/Hz); $N_{n.res}^2(f_s) = 10^{\frac{N_{o.n}^2(f_s)-30}{10}}$ – noise power spectral density at frequency f_s , W/Hz. Then:

$$\sigma_N^2(\Delta F_{res}) = \sigma_{n.a}^2(\Delta F_s) + \sigma_{n.res}^2(\Delta F_s). \quad (7)$$

Antenna noise power dispersion $\sigma_{n.a}^2(\Delta F_s)$ calculated as:

$$\sigma_{n.a}^2(\Delta F_s) = 10^{-12} 10^{\frac{E_{n.a}^*(f_s)}{10}} \Delta F_s h_a^2 / R_a, \quad (8)$$

where $E_{n.a}^*(f_s)$ – spectral sensitivity of the antenna at frequency f_s , dB (mV/m); $h_a = 2\sqrt{R_a G_a(f_s) \lambda^2 / 4\pi Z_0}$ – effective antenna height (m); R_a – antenna resistance (Ohm); $Z_0 = 120\pi$ – characteristic impedance of free space (Ohm); $G_a(f_s)$ – enemy antenna gain (dB).

As a result, expression (8) will be presented in the following form:

$$\sigma_{n.a}^2(\Delta F_s) = 9 \times 10^{\frac{E_{n.a}^*(f_s)+40}{10}} \Delta F_s G_a(f_s) / (\pi Z_0 f^2) \quad (9)$$

and the noise power dispersion can be calculated as (10). Then the formula for h^2 let's rewrite it to form (11).

$$\sigma_N^2(\Delta F_{res}) = \left(9 \times 10^{\frac{E_{n.a}^*(f_s)+40}{10}} G_a(f_s) / (\pi Z_0 f^2) + 10^{(N_{o.n}^2(f_s)-30)/10} \right) \Delta F_s. \quad (10)$$

$$h^2 = \frac{P_{tr} G_{tr}(f_s) G_a(f_s) \lambda^2}{(4\pi R)^2 \left(9 \times 10^{\frac{E_{n.a}^*(f_s)+40}{10}} G_a(f_s) / (\pi Z_0 f^2) + 10^{(N_{o.n}^2(f_s)-30)/10} \right) \Delta F_s}. \quad (11)$$

If the parameters of the radio monitoring receiver are not determined, then the calculation can be carried out based on the EIRP value equal to 38.2 dBW (this value is determined for an effective antenna diameter of 48 sm and a beam angle of 2.8 °, see Table 4). In formula (4), the value $G_a(f_s)$ corresponds to the gain of the antennas of the SC SSC Starlink terminal.

Taking into account recommendation ITU-R P.452-16 (Prediction procedure for assessing interference between stations located on the surface of the Earth at frequencies above approximately 0.1 GHz) [9], we use the formula for signal attenuation in free space at line-of-sight ranges:

$$L = 92.5 + 20\lg f + 20\lg R + Ag, \text{ dB}, \quad (12)$$

where Ag is the total absorption in atmospheric gases, which, according to Recommendation ITU-R P.676-10 (Attenuation in atmospheric gases) for the Ku-band is 0.05 dB/km [9].

Then we finally obtain for the frequency $f = 14$ GHz:

$$L = 92.5 + 20\lg f + 20\lg R + 0.07R, \text{ dB} \quad (13)$$

and for frequency $f = 14.5$ GHz:

$$L = 92.5 + 20\lg f + 20\lg R + 0.05R, \text{ dB}. \quad (14)$$

In formulas (13) and (14), the values of f are in (GHz), and the values of R are in (km). In Fig. 4 shows graphs of signal attenuation in free space under conditions of no rain $L1(R)$ for a radio link at a frequency of $f = 14$ GHz and $L2(R)$ for a radio link at a frequency of $f = 14.5$ GHz. The value of signal attenuation $L0$ at a distance of $R = 1$ km is also shown here.

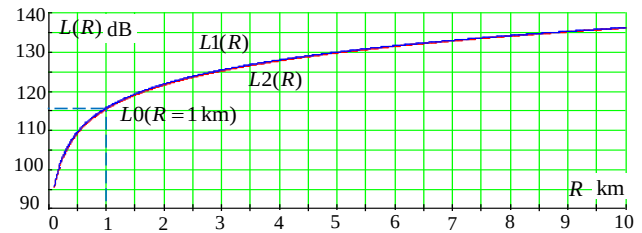


Fig. 4. Dependence of Attenuation of UT SSC Starlink Signals in Free Space

Taking into account the EIRP AT SSS Starlink, the final version of the signal attenuation value when it propagates freely is determined by the graph presented in Fig. 5.

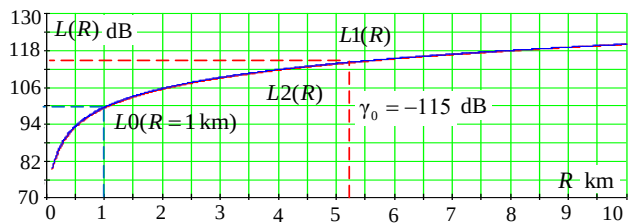


Fig. 5. Signal level of UT SSC Starlink Signals in Free Space as it Propagates Along the Main RP Lobe

Note that the graphs in Fig. 5 are constructed taking into account the propagation of the SSC Starlink UT signal in the direction of the main RP lobe. Analysis of the results obtained allows us to conclude that the level of signal attenuation is practically independent of frequency. And at a distance of 1 km it will reach a value of 100 dB.

At the same time, taking into account that the threshold sensitivity of the receiver γ_0 is determined by the expression [10]:

$$\gamma_0 \approx 10 P_N = 10(kT_2 \Delta f), \quad (15)$$

where P_N – thermal noise power at the receiver input; T_Σ – effective noise temperature of the antenna $\approx 410^\circ\text{K}$; $k = 1,23 \times 10^{-23}$, J/K – Boltzmann constant; $\Delta f = 60$ MHz – noise receiver bandwidth in which noise is measured.

According to expression (15), the threshold sensitivity of the receiver will be $\gamma_0 = -115$ dB.

Taking into account the obtained value, it is possible to determine the control range of ground-based UT SSC Starlink, equal to the order of $R_c = 5$ km (shown by the red dotted line in the graphs of Fig. 5). Note that the obtained values do not take into account the parameters of the antenna of the control complex. Thus, with a gain of even about 5 dB, the range R_c will increase to 10 km. But it should be taken into account that the calculations performed correspond to the location of the control complex along the main lobe of the RP UT SSC Starlink.

It should be noted that the calculated values were obtained for the condition that radio monitoring systems are located within the main radiation lobe, which in practice is quite difficult to ensure.

So, according to Table 4, the main lobe is a fairly limited area of space. In particular, in Fig. 6 shows the radiation pattern of the ground terminal antenna in the vertical plane.

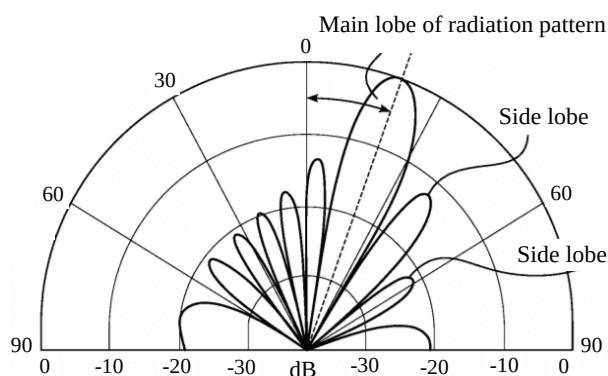


Fig. 6. Directional Pattern of the UT SSC Starlink Antenna

In the event that the guidance device does not ensure alignment of the radiation patterns within the main lobe of the antenna system of the radio monitoring receiver and the UT SSC Starlink, then, according to Fig. 6,

then electromagnetic accessibility will be reduced by 12–22 dB. In such conditions, receiving signals will become almost impossible.

In addition, the article somewhat idealizes the situation of temporary contact between the satellite and the ground terminal; according to [2], the contact time may be shorter. But these conditions equally negatively affect both the radio monitoring system and the subscriber terminal. Therefore, this situation is considered from the standpoint of general restrictions.

A more complex problem that developers of radio monitoring equipment may encounter is the quasi-random nature (for an outside observer) of switching the main lobe of the radiation pattern when changing the next spacecraft leaving the accessibility zone [2].

A solution to this problem will be addressed in a subsequent study.

Conclusion

The study showed that, despite the complex operating algorithm of SSC Starlink, based on handover operations (transfer of control over the Starlink SSC UT from one SC to another without disruption or loss of communication service), control of the operation of ground-based subscribers is theoretically possible. Because with a receiver sensitivity of -115 dB, the monitoring range is more than 5 km.

But it should be understood that in the Ku-band the level of signal attenuation along the Earth's surface is quite significant, so receiving monitoring equipment should be located on lifting platforms.

In addition, taking into account that in order to make contact between SC and UT, which is only 250 s, a constant change in the relative position of their RP is necessary, it is advisable to use widely directional antennas on control complexes. And although such a solution will reduce the efficiency of electromagnetic contact, it will significantly facilitate the practical implementation of antenna devices when they are placed on a flight-lifting platform.

Directions for further research will be related to the use of joint signal processing methods [11–13], in the interests of improving the quality of control.

References

1. Pehterev S.V., Makarenko S.I., Kovalsky A.A. Descriptive Model of Starlink Satellite Communication System. *Systems of Control, Communication and Security*. 2022;4:190–255. DOI:10.24412/2410-9916-2022-4-190-255
2. Makarenko S.I. Noise Immunity of Starlink User Terminals. *Systems of Control, Communication and Security*. 2023;2:81–101. DOI:10.24412/2410-9916-2023-2-81-101
3. Peterson W., Birdsall T., Fox W. The theory of signal detectability. *Transactions of the IRE Professional Group on Information Theory*. 1954;4(4):171–212. DOI:10.1109/TIT.1954.1057460
4. Dvornikov S.V. Time-Varying Process Detection Method Based on Spectrum Character Multiplication with an Automatic Identification Threshold. *Proceedings of Scientific Instrumentation* 2004;14(4):92–97.
5. Green D.M., Swets J.A. *Signal Detection Theory and Psychophysics*. New York: Wiley Publ.; 1966.
6. Dvornikov S.V., Pogorelov A.A., Voznyuk M.A., Ivanov R.V. Evaluation of simulation resistance of control channels with frequency modulation. *Information and Space*. 2016;1:32–35.
7. Fink L.M. *Theory of Discrete Signal Transmission*. Moscow: Sovetskoe radio Publ.; 1963. 576 p.
8. Makarenko S.I. Noise immunity of ground-based subscriber terminals of the Starlink satellite communication system. *Control, Communication and Security Systems*. 2023;2:81–101. DOI:10.24412/2410-9916-2023-2-81-101

9. Rec. ITU-R P.452-16 *Prediction procedure for the evaluation of interference between stations on the surface of the Earth at frequencies above about 0.1 GHz*. July 2015.
10. Tomovich R., Vukobratovich M. *Opšta teorija osetljivosti*. Beograd: Institut "Kirilo Savić" Publ.; 1969. 240 p.
11. Dvornikov S.V. Theoretical Foundations of The Synthesis of Bilinear Energy Distributions of Non-Stationary Processes in the Frequency-Temporary Space (Review). *Proceedings of Telecommun. Univ.* 2018;4(1):47–60.
12. Dvornikov S.V. Modification of Time-Frequency Descriptions of Non-Stationary Processes Based on Exponential and Power Functions. *Scientific Instrumentation*. 2004;14(3):76–85.
13. Dvornikov S.V., Kudryavtsev A.M. *Theoretical bases of frequency-time short-time signals analysis*. St. Petersburg: Telecommunication Military Academy Publ., 2010. 240 p.

Список источников

1. Пехтерев С.В., Макаренко С.И., Ковальский А.А. Описательная модель системы спутниковой связи Starlink // Системы управления, связи и безопасности. 2022. № 4. С. 190–255. DOI:10.24412/2410-9916-2022-4-190-255
2. Макаренко С.И. Помехозащищенность наземных абонентских терминалов системы спутниковой связи Starlink // Системы управления, связи и безопасности. 2023. № 2. С. 81–101. DOI:10.24412/2410-9916-2023-2-81-101
3. Peterson W., Birdsall T., Fox W. The theory of signal detectability // IRE Professional Group on Information Theory. 1954. Vol. 4. Iss. 4. PP. 171–212. DOI:10.1109/TIT.1954.1057460
4. Дворников С.В. Метод обнаружения на основе посимвольного перемножения реализаций спектра наблюдаемого процесса с автоматическим расчетом порога принятия решения // Научное приборостроение. 2004. Т. 14. № 4. С. 92–97.
5. Green D.M., Swets J.A. *Signal Detection Theory and Psychophysics*. New York: Wiley, 1966.
6. Дворников С.В., Погорелов А.А., Вознюк М.А., Иванов Р.В. Оценка имитостойкости каналов управления с частотной модуляцией // Информация и космос. 2016. № 1. С. 32–35.
7. Финк Л.М. Теория передачи дискретных сообщений. М.: Советское радио, 1963. 576 с.
8. Макаренко С.И. Помехозащищенность наземных абонентских терминалов системы спутниковой связи Starlink // Системы управления, связи и безопасности. 2023. № 2. С. 81–101. DOI:10.24412/2410-9916-2023-2-81-101
9. Рек. МСЭ-R P.452-16 (07/2015). Процедура прогнозирования для оценки помех между станциями, находящимися на поверхности Земли, на частотах выше приблизительно 0,1 ГГц.
10. Томович Р., Вукобратович М. Общая теория чувствительности. Пер. с сербск. М.: Советское радио, 1972. 240 с.
11. Дворников С.В. Теоретические основы синтеза билинейных распределений энергии нестационарных процессов в частотно-временном пространстве (обзор) // Труды учебных заведений связи. 2018. Т. 4. № 1. С. 47–60.
12. Дворников С.В., Сауков А.М. Модификация частотно-временных описаний нестационарных процессов на основе показательных и степенных функций // Научное приборостроение. 2004. Т. 14. № 3. С. 76–85.
13. Дворников С.В., Кудрявцев А.М. Теоретические основы частотно-временного анализа кратковременных сигналов. СПб.: ВАС, 2010. 240 с.

Статья поступила в редакцию 08.10.2023; одобрена после рецензирования 07.11.2023; принята к публикации 13.11.2023.

The article was submitted 08.10.2023; approved after reviewing 07.11.2023; accepted for publication 13.11.2023.

Информация об авторе:

ДВОРНИКОВ
Сергей Викторович

доктор технических наук, профессор, профессор кафедры радиотехнических и оптоэлектронных комплексов (Кафедра 21) Санкт-Петербургского государственного университета аэрокосмического приборостроения, профессор кафедры радиосвязи Военной академии связи имени Маршала Советского Союза С.М. Буденного

 <https://orcid.org/0000-0002-4889-0001>

Обзорная статья

УДК 004.7

DOI:10.31854/1813-324X-2023-9-6-42-57



Анализ методов идентификации трафика для управления ресурсами в SDN

Юлия Сергеевна Дмитриева ✉, dmitrieva@sut.ru
 Дарина Владимировна Окунева, okuneva.dv@sut.ru
 Василий Сергеевич Елагин, v.elagin@sut.ru

Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича,
Санкт-Петербург, 193232, Российская Федерация

Аннотация: Статья посвящена анализу методов классификации трафика в сети SDN. Выполнен обзор аналитических подходов идентификации трафика для выявления применяемых в них решений, а также оценки их применимости в сети SDN. Рассмотрены виды машинного обучения и выполнен анализ входных параметров. Методы интеллектуального анализа, освещенные в научных статьях, систематизированы по следующим критериям: параметры идентификации трафика, модель нейронной сети, точность идентификации. На основании анализа результатов обзора сделан вывод о возможности применения рассмотренных решений, а также о необходимости формирования схемы сети SDN с модулем элементов искусственного интеллекта для балансировки нагрузки.

Ключевые слова: Software-Defined Networking (SDN), Программно-конфигурируемая сеть (ПКС), Deep packet inspection (DPI), Machine learning (ML), Deep learning (DL), Convolutional Neural Network (CNN), искусственный интеллект (ИИ)

Финансирование: Публикация подготовлена в рамках прикладных научных исследований СПбГУТ, регистрационный номер 123060900012-6 в ЕГИСУ НИОКТР.

Ссылка для цитирования: Дмитриева Ю.С., Окунева Д.В., Елагин В.С. Анализ методов идентификации трафика для управления ресурсами в SDN // Труды учебных заведений связи. 2023. Т. 9. № 6. С. 42–57. DOI:10.31854/1813-324X-2023-9-6-42-57

Analyzing Traffic Identification Methods for Resource Management in SDN

Julia Dmitrieva ✉, dmitrieva@sut.ru
 Darina Okuneva, okuneva.dv@sut.ru
 Vasiliy Elagin, v.elagin@sut.ru

The Bonch-Bruevich Saint-Petersburg State University of Telecommunications,
St. Petersburg, 193232, Russian Federation

Abstract: The article is devoted to the analysis of traffic classification methods in SDN network. The review of analytical approaches of traffic identification to identify the solutions used in them, as well as assessing their applicability in the SDN network. Types of machine learning are considered and input parameters are analyzed. The methods of intelligent analysis covered in the scientific articles are systematized according to the following criteria: traffic identification parameters, neural network model, identification accuracy. Based on the analysis of the review results, the conclusion is made about the possibility of applying the considered solutions, as well as the need to form a scheme of SDN network with a module of artificial intelligence elements for load balancing.

Keywords: Software-Defined Networking (SDN), Deep packet inspection (DPI), Machine learning (ML), Deep learning (DL), Convolutional Neural Network (CNN), Artificial Intelligence (AI)

Funding: The scientific article was prepared in the framework of applied scientific research of SPBSUT, registration number 123060900012-6 in EGISU R&D.

For citation: Dmitrieva J., Okuneva D., Elagin V. Analyzing Traffic Identification Methods for Resource Management in SDN. *Proceedings of Telecommun. Univ.* 2023;9(6):42–57. DOI:10.31854/1813-324X-2023-9-6-42-57

Введение

По мере роста спроса на персонализированные сетевые услуги неизбежно возникает необходимость в детальной классификации сетевых потоков. Управление ресурсами в программно-конфигурируемой сети (SDN, аббр. от англ. Software-Defined Networking) является важной задачей, для которой существует большое количество подходов и методов [1–3]. В рамках самых перспективных методов балансировки трафика используется идентификация трафика для предобработки и его дальнейшего использования. Необходимо рассмотреть классификацию потоков для выбора наиболее эффективного метода управления ресурсами. В программно-конфигурируемой сети (ПКС) существует много подходов к управлению ресурсами, наиболее перспективные из них используют классификацию приложений, идентификацию трафика для повышения эффективности своей работы. Рассмотрим методы идентификации трафика, чтобы выявить наиболее подходящий и перспективный метод для управления ресурсами.

Актуальность и важность анализа сетевого трафика обусловлены ростом объема данных, сложностью сетевой инфраструктуры, угрозами безопасности и необходимостью оптимизации ресурсов [4–6].

В ряде задач управления ресурсами в SDN, затронутых в данной статье, невозможно рассматривать методы идентификации трафика отдельно от методов классификации и необходимо учитывать балансировку нагрузки.

Структура статьи представляет из себя следующее: в первом разделе рассматриваются методы идентификации трафика, во втором разделе – виды машинного обучения и проводится анализ входных параметров трафика для идентификации и обучения нейронной сети. В третьем разделе уделяется внимание балансировке нагрузки в SDN, основанной на машинном обучении.

Методы идентификации трафика

Методы идентификации сетевого трафика можно разделить на виды [7]: сигнатурный анализ, классификация на основе блоков данных, машинное обучение и другие. Рассмотрим каждый из методов подробнее.

Сигнатурный анализ – это метод анализа трафика, основанный на сравнении пакетов данных с заранее определенными сигнатурами. Сигнатуры – это уникальные шаблоны пакетов данных, которые соответствуют определенным типам трафика. Например, сигнатура веб-трафика будет содержать информацию о типе запроса (GET или POST), URL-адресе и т. д.

Сигнатуры представляют собой уникальные характеристики или паттерны, которые характерны для определенного типа трафика и могут быть основаны на известных характеристиках протоколов, параметрах портов, последовательностях байтов и других факторах, связанных с определенными видами трафика или атак. Однако этот метод также имеет свои ограничения, например, он может быть неэффективен при обнаружении новых типов трафика. Поэтому, помимо анализа сигнатур, также важно использовать другие методы и подходы для классификации трафика [8].

Для формализации различных подходов сигнатурного анализа авторами предложены следующие соотношения (1-4), используемые для идентификации трафика:

1) анализ шаблонов – заключается в сравнении эталонных образцов пакетов конкретных приложений (шаблонов) с пакетами, передаваемыми в исследуемом потоке трафика (1), где $[e_x \dots e_y]$ – последовательность элементов (поля пакета), S – последовательность переданных элементов (поля пакета), C_z – эталонная подпоследовательность, $Class^c$ – набор классов;

2) анализ протокола/состояния – состоит в определении состояний сетевого потока с использованием соответствующего графа, который представляет собой заранее определенный граф состояний для конкретных приложений или протоколов, позволяющий по последовательности прохождения узлов однозначно производить идентификацию (2), где S – последовательность переданных элементов (поля пакета), $[e_i]$ – элемент (поле пакета); $Index_{e_{k-1}}, Index_{e_k}(S)$ – индекс элемента e_{k-1}, e_k ;

3) эвристический и поведенческий анализ – заключается в поиске заданной последовательности пакетов в потоке трафика (3), где S – последовательность переданных элементов (пакеты), $[e_1 \dots e_N]$ – последовательность элементов (характеристика пакета), T – цель или последний узел графа,

$Model\ Class^c([e_x \dots e_y])$ – процесс моделирования модели, который позволяет проходить все узлы графа $e_x \dots e_y$, e_t – узел, принадлежащий T^c .

4) числовой анализ – вычисляет характерные для отдельных приложений параметры пакетов в соответствующей последовательности и сравнивает их с эталонными (4), где x – индекс пакета, p – индекс характеристики, P_p – характеристика пакета e_x с индексом p , P_p – набор параметров пакета, V_p^c – множество значений, которым принадлежит пакет (множество значений для пакетов), $V_{P_k}^c$ – множество значений сигнатур, принадлежащих какому-либо классу, т. е. для любого k из набора параметров пакета P_k принадлежит множеству $V_{P_k}^c$.

В сигнатурном анализе выделен комплексный подход к классификации трафика – метод глубокого анализа пакетов (DPI, аббр. от англ. Deep Packet Inspection), основанный на предварительно вычисленных или самообучающихся сигнатурах. Он позволяет проводить глубокий анализ пакетов данных, передаваемых по сети, и выявлять различные типы трафика, такие как веб-серфинг, потоковое видео, файлы, мессенджеры и т. д. Преимуществом сигнатурного анализа является быстрота обработки трафика и высокая точность выявления типов трафика. Однако этот метод неэффективен для обнаружения новых типов трафика или для обхода сетевых устройств, которые могут изменять пакеты данных.

$$\begin{cases} S \equiv [e_1 \dots e_n] \\ \{Class^c\}: [C_z]^c \Rightarrow Class^c \\ Method_{pa}(S, \{Class^c\}) = \exists_{x,y}: [e_x \dots e_y] \equiv [C_z]^c \Rightarrow S \in Class^c \end{cases} \quad (1)$$

$$\begin{cases} S = [e_i] \\ \{Class^c\}: Class^c \Rightarrow [e_k]^c \\ Method_{p/sa}(S, \{Class^c\}) = \exists [e_k] = [e_k]^c, e_k \in S: Index_{e_{k-1}}(S) < Index_{e_k}(S) \rightarrow S \in Class^c \end{cases} \quad (2)$$

$$\begin{cases} S \equiv [e_1 \dots e_N] \\ Method_{pea}(S, \{Class^c\}) = \exists_{x,y}: ModelClass^c([e_x \dots e_y]) \rightarrow e_t \in T^c \Rightarrow S \in Class^c \end{cases} \quad (3)$$

$$\begin{cases} S \equiv [e_1 \dots e_x \dots e_y \dots e_k] \\ Method_{na}(S, \{Class^c\}) = \exists_x, \exists_p: P_{P_k}(e_x) \in V_{P_k}^c \Rightarrow S \in Class^c \end{cases} \quad (4)$$

Классификация на основе блоков данных базируется на анализе открытых полей пакета, таких как порты, IP-адрес отправителя/получателя или MAC отправителя/получателя и т. д. Данный метод является наиболее применимым, но он не работает с зашифрованным и туннелированным трафиком.

До появления DPI и машинного обучения, сопоставление типов портов преимущественно использовалось для идентификации сетевого трафика, но в настоящее время этот метод для таких целей не используется, в основном, соответствует простой концепции обнаружения трафика; большинство приложений P2P (аббр. от англ. Peer to Peer, одноранговая сеть) имеют свой порт по умолчанию, например, BitTorrent имеет порт 6881 6889 TCP/UDP. Если у специалиста нет дорогостоящей системы обнаружения вторжений, то без использования каких-либо идентификаторов сетевой администратор может определить тип сетевого трафика. Сейчас P2P-приложения используют динамический порт, в этом случае сопоставление типов портов неэффективно, а также невозможно правильно идентифицировать сетевой трафик и P2P-приложение.

Машинное обучение – это метод анализа трафика, основанный на использовании алгоритмов машинного обучения для выявления типов трафика. Алгоритмы машинного обучения обучаются на основе большого количества примеров пакетов данных различных типов, и затем используются для классификации новых пакетов данных. Преимуществом машинного обучения является его способность обнаруживать новые типы трафика и обходить сетевые устройства, которые изменяют пакеты данных. Кроме того, этот метод более точен, чем анализ поведения. Однако машинное обучение требует больших вычислительных ресурсов для обработки трафика и обучения алгоритмов.

В зависимости от конкретных задач и требований, в системах DPI могут использоваться различные методы анализа трафика. Комбинация нескольких методов может повысить эффективность обнаружения типов трафика и уменьшить количество ложных срабатываний.

Самым популярным методом исследования пакетов в сети на данный момент является DPI. Однако система глубокого анализа пакетов имеет ряд недостатков: большая нагрузка на сеть, невозможность анализировать зашифрованный трафик и низ-

кая производительность в работе с новыми видами приложений. Таким образом, система глубокого анализа пакетов теряет свою эффективность в связи с большим ростом трафика в сети Интернет и с резким увеличением количества различных сервисов и приложений. Системы искусственного интеллекта (ИИ) по сравнению с системой DPI более производительны, гибки за счет самостоятельного обучения и лишены вышеуказанных недостатков.

Наиболее часто используемые методы классификации трафика – по типу портов и на основе анализа полезной нагрузки, имеют ряд недостатков и не позволяют классифицировать приложения нового поколения, такие как P2P и онлайн-игры [9], а точность этого метода составляет не более 30–70 % [10].

Метод анализа полезной нагрузки, представляющий собой глубокую проверку сетевых пакетов, дает хорошие результаты, особенно для P2P-сетей, но его нельзя использовать для зашифрованного трафика, и одной из проблем этого метода является отсутствие конфиденциальности пользователей, которая не соблюдается из-за углубленных проверок трафика [11]. Чтобы преодолеть ограничения предыдущих методов и повысить точность классификации трафика, принято решение использовать алгоритмы машинного обучения [12–14].

Виды машинного обучения

Существуют разные виды искусственного интеллекта, из которых самым известным и широко используемым является машинное обучение (ML, аббр. от англ. Machine Learning). Основное преимущество применения методов машинного обучения заключается в сокращении времени обработки больших объемов данных и высокой точности результатов.

Машинное обучение позволяет решать ряд задач, среди которых можно выделить задачу классификации трафика, состоящую из трех основных этапов.

На первом этапе происходит обучение модели на основе обучающего набора данных. Обучающий набор данных содержит объекты трафика, представленные в виде векторов признаков, которые помогают модели выделить определенный целевой класс. Кроме того, обучающий набор данных также содержит метки классов, которые указывают, к какому классу каждый объект трафика относится.

На втором этапе происходит проверка работы модели на тестовом наборе данных. Тестовый набор данных представляет собой независимую выборку, которая не использовалась в процессе обучения. Модель применяется к тестовым данным, и их классификация сравнивается с известными метками классов. Это позволяет оценить точность и производительность модели.

На третьем этапе модель применяется к новым данным для определения их принадлежности к целевому классу. Новые данные представляют собой потоки сетевого трафика, представленные последовательностями IP-пакетов. Данные характеризуются различными признаками: количеством пакетов, IP-адресами, номерами портов и статистическими особенностями потоков. Модель использует полученные знания о зависимостях между признаками и классами, чтобы определить, к какому классу относится каждый новый объект трафика.

Таким образом, процесс классификации трафика методами машинного обучения включает обучение модели на обучающем наборе данных, проверку работы модели на тестовом наборе данных и использование модели для классификации новых данных в целевой класс.

Рассмотрим основные виды машинного обучения, которые используются для классификации сетевого трафика:

1) логистическая регрессия используется для бинарной классификации, то есть разделения объектов на два класса (применяется для прогнозирования вероятности принадлежности объекта к определенному классу на основе признаков);

2) деревья решений представляют собой структуру, состоящую из решающих правил, которые помогают принять решение о классификации объекта (основываются на иерархическом разбиении признакового пространства);

3) случайный лес – это ансамбль деревьев решений, где каждое дерево обучается на случайной подвыборке данных и случайном подмножестве признаков, т. е. каждое дерево принимает решение независимо, а итоговое решение определяется голосованием или усреднением результатов всех деревьев;

4) метод опорных векторов (SVM, аббр. от англ. Support Vector Machine) осуществляет поиск оптимальной разделяющей гиперплоскости между различными классами (строит модель, которая классифицирует новые данные, исходя из их положения относительно этой гиперплоскости);

5) нейронные сети состоят из набора обрабатывающих нейронов, которые взаимосвязаны и преобразуют набор входных данных в набор требуемых выходов (результат преобразования распознается по характеристикам нейронов и воздействию, связанному с корреляцией между ними, за счет улучшения связей между узлами сеть может получать необходимые результаты);

6) k -ближайших соседей (k -NN, аббр. от англ. k -Nearest Neighbors) – это метод классификации, основанный на близости объектов, классифицирует новый объект, исходя из классов его k ближайших соседей в обучающем наборе данных;

7) градиентный бустинг – это ансамблевый метод, который комбинирует несколько слабых моделей (обычно деревьев решений) для создания более мощной модели, обучает модели последовательно, взвешивая ошибки предыдущих моделей и пытаясь исправить их;

8) статистический подход опирается на статистические характеристики для идентификации приложения и определяет, какие приложения создают определенные потоки трафика, в частности, Байесовский классификатор – ML-алгоритм, предназначенный для многоклассовой классификации данных с независимыми признаками [15].

В работе [16] предложено несколько типов методов интеллектуального анализа данных классификации:

- наивный Байес (*от англ. Naive Bayes*);
- J48 (Java-реализация алгоритма C4.5 с открытым исходным кодом в программном обеспечении WEKA (*аббр. от англ. Waikato Environment for Knowledge Analysis*);
- проективная адаптивная резонансная теория (PART, *аббр. от англ. Projective Adaptive Resonance Theory*);
- сетевой алгоритм радиально-базисных функций (RBF, *аббр. от англ. Radial Basis Function*) для обнаружения вторжений.

Авторы также собрали набор данных из KDD-cup [17] (*аббр. от англ. Knowledge Discovery and Data Mining Tools*), который имеет 24 типа атак. Сравнивая выше перечисленные методы, отмечено, что алгоритм PART показывает лучшие результаты обнаружения вторжений.

Выбор характеристик потока трафика для обучения модели

Классификация трафика может производиться по различным параметрам, включая тип протокола, порты назначения и источника, размер пакета, время отправки и приема пакета, количество пакетов в потоке и т. д.

Сейчас в большинстве систем классификация производится для потоков. Поток идентифицируется по характеристикам Five-Tuple:

- IP-адрес источника;
- порт источника;
- IP-адрес получателя;
- порт получателя;
- протокол транспортного уровня.

По этим пяти составляющим определяется принадлежность пакета к тому или иному потоку. Таким образом, классификация трафика производится для потока, а не для каждого пакета. В потоке выбирается N число пакетов, которое используется для классификации, затем весь поток может быть идентифицирован как приложение, которое было распознано в N пакетах.

Входные параметры для модели машинного обучения

Тип протокола прикладного уровня, определяет формат и способы обмена данными между приложениями в компьютерной сети. Приложение использует определенный протокол для обмена данными. Например, HTTP используется для передачи веб-страниц и медиа-контента, SMTP – для отправки и получения электронной почты, FTP – для передачи файлов, а SSH – для удаленного управления устройствами.

Порт источника и назначения (порт – это числовой идентификатор, который используется на транспортном уровне модели OSI для определения, к какому протоколу относится пакет). Хотя порты часто связываются с конкретными приложениями, нельзя гарантировать, что порт всегда указывает на определенный тип приложения, однако, существуют их устойчивые комбинации.

IP-адрес источника и IP-адрес назначения. Часто приложения, такие как Google, Whatsapp, Amazon, Twitter, имеют свои серверы. Соответственно, по IP-адресу получателя в Uplink потоке можно определить, к какому ресурсу был направлен запрос.

Использование протокола шифрования. Большая часть приложений в сети шифруется. Это важно для обеспечения конфиденциальности, целостности и аутентификации данных. Наиболее распространенные протоколы шифрования – это SSL (*аббр. от англ. Secure Sockets Layer*, уровень защищенных сокетов) и TLS (*аббр. от англ. Transport Layer Security*, протокол защиты транспортного уровня). Они используются для шифрования трафика во многих приложениях, особенно тех, которые требуют высокой степени безопасности и защиты данных.

Размер пакета в байтах – является одним из основных параметров сетевого трафика, который может отличаться в зависимости от приложения. Например, мессенджеры могут передавать текстовые сообщения меньшего размера, в то время как ВКС (видеоконференцсвязь) может передавать аудио- и видеоданные большего размера. Этот параметр может быть изменен с помощью сжатия.

Количество переданных пакетов в секунду – параметр, который может показать интенсивность использования приложения. Например, при использовании мессенджера количество переданных пакетов в секунду будет ниже, чем при использовании ВКС. Этот параметр может быть изменен с помощью установки ограничений на скорость передачи данных.

Задержка передачи данных – параметр, который может отличаться в зависимости от типа приложения. Например, при использовании мессенджера

задержка передачи данных будет ниже, чем при использовании ВКС.

Длительность потока – в сети это время от начала потока (инициирования сессии доступа к ресурсу) и до его окончания. Разные типы приложений обычно генерируют трафик с разной длительностью потока. Например, потоки, связанные с веб-браузерами, обычно более короткие, потоки медиаплееров – более длительные, потоки файловых менеджеров имеют разную длительность в зависимости от размера передаваемых файлов.

Количество потоков передачи данных – было выбрано как параметр, который может показать количество одновременных соединений с сервером. Например, при использовании мессенджера количество потоков передачи данных будет ниже, чем при использовании ВКС.

Количество пакетов в потоке – например, для видео и аудиопотоков характерно большое количество пакетов в секунду, причем их размер является постоянной величиной. В то же время приложения, связанные с передачей файлов, могут иметь большое количество пакетов с разным размером.

Длина заголовка. Каждый протокол имеет свой уникальный формат заголовка пакета, длина заголовка может быть разной для разных протоколов. Например, заголовок протокола TCP (*аббр. от англ. Transmission Control Protocol*, протокол управления передачей) имеет длину 20 байт, а заголовок протокола UDP (*аббр. от англ. User Datagram Protocol*, протокол пользовательских дейтаграмм) имеет длину 8 байт.

Сигнатуры – набор данных (условий), которые необходимо найти в трафике, проверяют большое количество полей различных протоколов, таких, как исходный адрес, порт назначения или TCP флаги.

Время поступления данных – моменты времени поступления пакетов на приемной стороне, в некоторых случаях, время начала сессии.

Незашифрованные пакеты подтверждения – пакеты, служащие подтверждающими сообщениями для различных протоколов.

Идентификатор протокола – элемент в структуре IP-пакета, следующий за заголовком протокола, занимает один байт и указывает, какому протоколу верхнего уровня принадлежит информация, размещенная в поле данных пакета.

Пакеты данных для доставки и обслуживания сети – трафик служебных протоколов: ICMP (*аббр. от англ. Internet Control Message Protocol*, протокол межсетевых управляющих сообщений), DHCP (*аббр. от англ. Dynamic Host Configuration Protocol*, протокол динамической настройки узла).

Временные интервалы между пакетами.

Для ML-моделей и нейронных сетей используют разные комбинации входных параметров в зависимости от поставленной задачи, типа сети ИИ, необходимой точности классификации трафика. Комбинация нескольких параметров может улучшить точность классификации сетевого трафика.

Некоторые из таких параметров, как порты и протоколы, могут быть недоступны для анализа в случае зашифрованного трафика. Это связано с тем, что в этом случае данные отправляются в зашифрованном виде, что делает невозможным идентификацию протоколов в обычном виде. Тогда для анализа зашифрованного трафика могут быть использованы другие параметры, в частности, шаблоны трафика, размер пакетов, частота и время передачи данных.

Согласно [18], только от 30 до 70 % текущего интернет-трафика может быть классифицировано с использованием методов на основе анализа типов портов. Поэтому для классификации современных сетевых услуг необходимы более сложные методы идентификации.

Традиционные методы классификации создают ряд проблем для достижения более высокой точности. Во-первых, классификация на основе параметров портов не обеспечивает достаточную точность, когда в сетевой системе используется переназначение, переадресация и случайное назначение портов. Во-вторых, подход, основанный на сигнатуре полезной нагрузки, плохо работает для зашифрованных пакетов. В-третьих, подходы, основанные на статистике, весьма чувствительны к динамически изменяющимся условиям сети и приложений, таким как уровень загруженности сети, интенсивность перекрестного трафика и поведение, зависящее от пользователя. Предлагается более обстоятельно рассмотреть альтернативные способы идентификации трафика на основе применения машинного обучения.

Нейронная сеть анализирует информацию и позволяет оценить согласованность анализируемых данных с характеристиками, которые она научена распознавать.

В работе [19] авторы в качестве модели сети выбрали многослойную нейронную сеть прямого пространства. Исследуется эффективность применения нейронной сети для диагностики аномальной сетевой активности. Искусственно моделируется активность сети путем сканирования портов локального компьютера с удаленной машины. Полученные данные представляют собой выборку для обучения нейронной сети и определяют условия протекания нормальной сетевой активности и возникновения аномальной. Нейронная сеть классифицирует входное пространство признаков на два класса сетевой активности: нормальная (0) и аномальная (1).

Исследователи в работе [22] разработали искусственную модель выборки трафика по следующим параметрам:

- тип протокола транспортного уровня;
- номер назначения транспортного порта;
- сигнатура протокола, которая соответствует приложению, работающему с данным портом транспортного уровня.

Для решения задач идентификации и классификации сетевого трафика проведен сравнительный анализ трех основных типов нейронных сетей.

В докладе [24] на Международной конференции представлен фреймворк нейронной сети, который позволяет строить многоклассовые модели сетевого трафика, подходящие для задач генерации потоков и классификации.

В исследовании [26] решается задача мелкогранулярной классификации для потоков чата методом, основанном на сверточной нейронной сети (CNN, аббр. от *англ.* Convolutional Neural Network) для детальной классификации потоков чатов в реальном времени. Методы классификации сетевого трафика, рассмотренные в этой статье, сосредоточены на классификации зашифрованных потоков, в которых известны для каждого сетевого потока только данные из пяти кортежей: время поступления данных, протокол передачи по сети, размер пакета (при условии, что к конкретному содержимому пакету невозможно получить доступ), данные адреса источника и назначения передачи данных.

Авторы работы [29] на Международной конференции предложили новую классификацию зашифрованного трафика, основанную на полезной нагрузке, использующую пакеты подтверждения, которыми обмениваются конечные хосты для установления защищенной связи до того, как будут доставлены зашифрованные потоки данных.

Методы выбора признаков для классификации трафика производятся иранскими учеными в работе [30]. Они разработали фреймворк, который включает в себя две модели нейронной сети: Naive Bayes и SVM.

В [31] исследовали методы машинного обучения для идентификации приложений с помощью классификации сетевого трафика. В качестве классификаторов использовались сети Байеса.

В работе [32] изучается внедрение глубокой нейронной сети (DNN, аббр. от *англ.* Deep Neural Network) для классификации данных сетевого трафика, где DNN используется для автоматической классификации данных сетевого трафика, собранных с контроллера ONOS (аббр. от *англ.* Open Network Operating System) сети SDN. Исследователи подтвердили, что DNN в этом случае должна учитывать не только пакеты данных, предназначенные

для доставки, но и пакеты данных, необходимые для поддержки сети в рабочем состоянии, так как производительность классификации DNN зависит от данных сетевого трафика.

В исследовании [6] авторы используют методы машинного обучения в процессе идентификации приложений для смартфонов. Они рассматривают, как поведение трафика приложений меняется со временем на разных устройствах и в разных версиях приложений, и могут ли приложения для смартфонов быть идентифицированы путем анализа исходящего от них зашифрованного сетевого трафика.

Исследователи в статье [33] предлагают сквозной метод классификации зашифрованного трафика с использованием одномерных сверточных нейронных сетей (1D-CNN, аббр. от *англ.* One-Dimensional Convolution Neural Networks) [34]. Метод основан на глубоком обучении. 1D-CNN применяется в качестве алгоритма обучения и используется для автоматического изучения характеристик необработанного трафика. Объединяет разработку, извлечение и выбор признаков в единый фреймворк.

В работе [35] авторы занимались проблемой классификации трафика, которая заключалась в том, что признаки должны быть выделены экспертом. Поиск признаков, которые позволяют классифицировать трафик, очень утомителен и отнимает много времени, поэтому в данной работе входные параметры трафика выбираются с применением метода на основе оценочной модели, используется комбинация CNN, метаэвристического алгоритма ant-lion (ALO, аббр. от *англ.* Ant-Lion Optimization) и самоорганизующейся карты (SOM, аббр. от *англ.* Self Organizing Map) для создания модели классификации трафика, которая может точно идентифицировать типы трафика.

Основные этапы анализируемого метода:

- предварительная обработка данных о трафике с использованием энтропии и дисперсии энтропии;
- автоматическое извлечение признаков из скрытых слоев одномерной сверточной нейронной сети (1D CNN);
- эффективный выбор признаков с высокой точностью классификации с использованием ALO;
- классификация новых экземпляров с использованием кластеризации на основе fuzzy-SOM (*от англ.* fuzzy, нечеткая).

Предлагаемый метод позволяет идентифицировать зашифрованный трафик, динамические протоколы, такие как P2P, различать VPN-трафик и не являющийся VPN, классифицировать различные типы трафика с высокой точностью. Входные параметры трафика для идентификации и обучения нейронной сети представлены в таблице 2.

ТАБЛИЦА 2. Сравнение входных параметров трафика

TABLE 1. Comparison of Input Traffic Parameters

№	Ссылки	Модель нейронной сети	Параметр трафика	Точность идентификации, %
1	[19], [20], [21]	Многослойная нейронная сеть прямого распространения	– Тип протокола (TCP, UDP и др.) – Порт источника/порт назначения – Размер пакета	96,5
2	[22], [23]	Персептрон, многослойный персептрон, сеть со встречным распространением	– Тип протокола транспортного уровня – Номер назначения порта транспортного уровня – Сигнатура протокола, соответствующая приложению, работающему с данным протоколом	97
3	[24], [25]	Авторегрессионная	– Интервалы между пакетами – Размер пакета	99
4	[26], [27], [28]	CNN	– IP-адрес источника и IP-адрес назначения передачи – Время поступления данных – Протокол передачи – Размер пакета – Порт	94
5	[29], [30], [31], [32]	Байесовская нейронная сеть	– IP-адрес источника и IP-адрес назначения – Номера порта источника, номера порта назначения – Идентификатор протокола – Пакеты	99
6	[33], [34]	Одномерная CNN	Зашифрованный поток трафика	90,5
7	[35]	CNN метаэвристического алгоритма ant-lion и SOM	Параметры комбинируются индивидуально с использованием метода на основе оценочной модели	98

Исходя из анализа таблицы 2, по точности идентификации охватываемых параметров трафика представляется актуальным использовать модель CNN, метаэвристический алгоритм ant-lion, и ML-метод SOM, так как параметры комбинируются индивидуально с использованием метода на основе оценочной модели.

С учетом того, что основным элементом управления является ИИ, необходимо сформировать систему на базе ИИ, которая будет регулировать и управлять балансировкой трафика. Исходя из общей структуры SDN, модуль ИИ будет расположен между уровнем управления и уровнем данных, как показано на рисунке 1.

Балансировка нагрузки в SDN на основе искусственного интеллекта

Подходы к балансировке нагрузки в программно-конфигурируемой сети, основанные на машинном обучении, улучшают возможности обучения и позволяют системе принимать решения.

На рисунке 1 представлена предлагаемая схема структуры SDN, которая состоит из пяти уровней. Уровень модуля ИИ отвечает за производство, объединение и распространение правил и политик по сети.

Модуль ИИ позволяет сетевым менеджерам устранять проблемы до их возникновения внутри сети, увеличивать функциональность сети в ответ на меняющиеся требования и принимать превентивные меры для снижения рисков. Данные и/или правила, сгенерированные в модуле ИИ, используются для выявления и устранения сетевых проблем, настройки сети и т. д.

Уровень модуля ИИ состоит из трех подуровней: обработки данных, системы сбора и хранения исходных данных и подуровня генерации команд. Используя данные и информацию, уровень модуля ИИ создает правила и политики, применяя методы, основанные на алгоритмах или искусственном интеллекте. Собранные данные используются для построения правил посредством интеграции с намерениями пользователя.

Уровень обработки данных создает инновационные правила путем сравнения требований приложений с объединенными полученными знаниями о состоянии сети [36]. Принятие решений реализуется как модель, основанная на правилах, с использованием компьютерного языка Java или Lisp [37]. Правила, созданные на подуровне принятия решений, написаны на универсальном языке, чтобы другие уровни могли их понять. Для написания созданных правил используются специализированные языки правил, например, язык разметки правил (RuleML) [38], формат обмена правилами (RIF) [39], язык правил семантической сети (SWRL) [40].

Уровень системы сбора и хранения исходных данных включает общую базу данных для хранения правил, использует протоколы/языки для ввода, изменения, устранения правил/знаний и совместного использования [41]. Таким образом, рекомендации, полученные с помощью модели управления генерации правил и команд, формирование наборов данных и генерация значений данных, сетевые данные и сообщения с уровня управления составляют общую базу данных на этом подуровне.

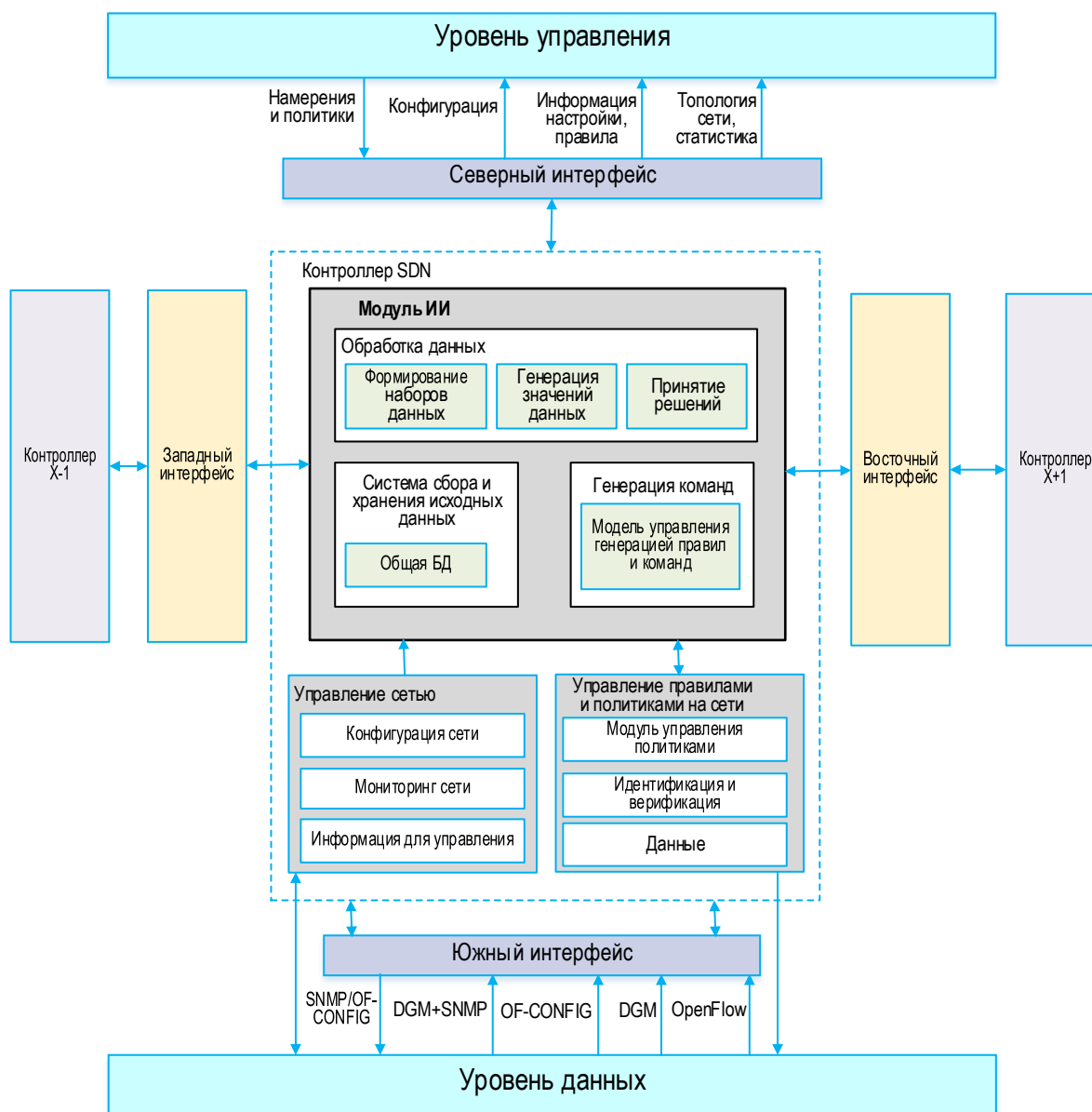


Рис. 1. Предлагаемая структура SDN с модулем ИИ

Fig. 1. Proposed Structure of SDN With AI Module

Механизм управления правилами и политиками на сети применяет правила или делает выводы на основе принятых решений, а затем дает указания в зависимости от того, насколько хорошо были применены правила и знания. Наиболее часто используемым протоколом настройки и управления сетью является протокол сетевой конфигурации (NETCONF, аббр. от англ. NETwork-CONFIguration Protocol). Благодаря совместимости с устройствами пересылки протокол OF-CONFIG (аббр. от англ. OpenFlow management-and-CONFIguration Protocol) передает данные через NETCONF [42].

OF-CONFIG или NETCONF могут быть заменены простым протоколом сетевого администрирования (SNMP, аббр. от англ. Simple-Network-Management Protocol), разработанным для наблюдения и настройки сетевых устройств в SDN [43].

Следует отметить, что подуровень управления сетью с помощью протокола SNMP/OF-CONFIG собирает данные конфигурации для настройки сети и сетевого наблюдения (статистика трафика, расположение сети, показатели производительности и т. д.). Другие данные могут быть собраны с использованием метода сбора данных DGM (аббр. от англ. Data Gathering Method), такого как оптимизация на основе квадратичного целочисленного программирования [44], выборка пакетов [45], адаптивный сбор данных [46] и сбор данных измерений датчиков [47]. Кроме того, уровень управления может собирать различные данные, включая расположение, настройку, схемы движения, записи о событиях, потреблении ресурсов, показатели производительности, показания датчиков и т. д.

Сетевые данные, необходимые для наблюдения за сетью на уровне управления и создания знаний в модуле элементов ИИ, хранятся в модуле, содержащим информацию для управления. Данные внутри подуровня информации для управления могут быть представлены с использованием языка представления данных YANG (*аббр. от англ. Yet Another Next-Generation*) [48] или CIM (*аббр. от англ. Common-Information Model, общая информационная модель*) [49].

Уровень управления и контроллер SDN взаимодействуют друг с другом благодаря «северному» интерфейсу, который может быть реализован с использованием *ad hoc* [50], RESTful (*от англ. Representational State Transfer, передача репрезентативного состояния*) [51], и сети на основе намерений IBN (*аббр. от англ. Intent-Based Networking*) [52, 53] или языкового API (*от англ. Application Programming Interface, прикладной программный интерфейс*). «Южный» интерфейс служит мостом между уровнем инфраструктуры и уровнем управления при использовании протоколов: OpenFlow, ForCES, OpFlex и других. Он используется для передачи необработанных данных от компонентов пересылки к контроллеру и для передачи правил потока от уровня управления к оборудованию уровня инфраструктуры. Используются интерфейсы, обеспечивающие связь между физически разнесенными контроллерами, ориентированными «с востока на запад»: ALTO [54], Hyperflow [55], ONOS [56], Onix [57] и т. д.

Реактивные подходы к управлению вызывают изменения в сети в ответ на потоки или события. Контроллер предварительно вычисляет компоненты коммутации с помощью набора правил при использовании проактивного управления для управления всеми потенциальными потоками трафика еще до того, как трафик достигнет коммутаторов.

Одной из основных операций контроллера является определение идеального маршрута передачи данных потока (вычисление пути) и оптимизация трафика, которая включает в себя улучшение потоков трафика с целью повышения эффективности сети с помощью собранных данных. Кроме того, уровень управления может собирать необработанные данные: информация о трафике, данные QoS (*аббр. от англ. Quality of Service, качество обслуживания*), правила, инциденты безопасности, протоколы, используемые для маршрутизации, и т. д. [58]. Более того, реализуя механизм политик, уровень управления может инструктировать сетевые устройства, выполнять конкретные задачи для удовлетворения конкретных требований, выполняя политики, как показано на рисунке 1. Сетевые рекомендации преобразуются в правила с помощью подуровня управления правилами и полити-

ками на сети путем учета дополнительной информации, других правил и представлений, основанных на знаниях [59].

Сетевые данные маршрутизируются через последовательность сетевых сервисов с использованием гибкой цепочки, где контроллер выбирает их для включения в цепочку в зависимости от изменяющихся сетевых условий [60]. Кроме того, контроллер также может гибко создавать частные сети, такие как частные виртуальные сети, и масштабировать их в зависимости от требований динамической сети [61].

Разработчики приложений используют прикладной уровень в качестве основы для передачи своих запросов в базовую физическую сеть. Кроме того, это позволяет сетевым менеджерам централизованно устанавливать руководящие принципы настройки сети, которые лучше соответствуют общим бизнес-целям и намерениям в сети IBN [53], при этом функции приложения отделены от аппаратного обеспечения, и определять сетевые политики, уникальные для приложений. Более того, рекомендации по применению могут постоянно изменяться в SDN в зависимости от информации о функционировании сети, что повышает качество предоставления услуг [62].

Прикладной уровень отделяет сервисную функцию от физических компонентов, чтобы централизованно определять желаемые намерения и правила. В результате, при изменении состояния сети принципы и намерения применения могут динамически изменяться. Для достижения задач по определению политики и обновления существуют фреймворки программирования, такие как Procera [63], Nettle [64], Frenetic [65], Kinetic [66] и т. д., которые построены поверх распространенных языков программирования: Python, Haskell и т. д.

В ближайшей перспективе наиболее актуальными прикладными направлениями применения модуля ИИ являются: оптимизация трафика (интеллектуальная пересылка пакетов, оптимизация нагрузки, поддержание и настройка QoS); автономное сетевое администрирование (администрирование действий пользователей; управление мобильностью).

Выводы

На основе обзора научных работ в статье были исследованы аналитические методы классификации трафика и модели машинного обучения на предмет анализа методов идентификации трафика для эффективного управления ресурсами в SDN.

Анализ методов идентификации трафика позволил сделать вывод о том, что аналитические методы классификации трафика имеют ограничения в применении, поэтому принято решение использовать алгоритмы машинного обучения. Был выполнен

анализ входных параметров трафика для идентификации и обучения нейронной сети, который показал, что по точности идентификации охватываемых параметров трафика представляется актуальным использовать CNN-модель, метаэвристический алгоритм ant-lion и ML-метод SOM, так как параметры комбинируются индивидуально с использованием метода на основе оценочной модели.

В статье определено место модуля ИИ в структуре SDN, с учетом особенностей построения таких сетей.

Идентификация трафика в сети SDN применяется для балансировки нагрузки, поддержания QoS и эффективного использования сетевого ресурса. Предложена структура внедрения модуля ИИ, в том числе, для идентификации трафика.

В будущих исследованиях необходимо проработать интерфейсы подключения модуля ИИ к SDN, для минимизации его влияния на задержку при сетевом обмене.

Список источников

1. Дмитриева Ю.С. Сравнительный анализ методов управления сетевыми ресурсами в сетях SDN // Труды учебных заведений связи. 2022. Т. 8. № 1. С. 78–83. DOI:10.31854/1813-324X-2022-8-1-73-83
2. Kirichek R., Vladyko A., Zakharov M., Koucheryavy A. Model networks for Internet of Things and SDN // Proceedings of the 18th International Conference on Advanced Communication Technology (ICACT, PyeongChang, Korea (South), 31 January 2016 – 03 February 2016). IEEE, 2016. PP. 76–79. DOI:10.1109/ICACT.2016.7423280
3. Muhizi S., Shamshin G., Muthanna A., Kirichek R., Vladyko A., Koucheryavy A. Analysis and Performance Evaluation of SDN Queue Model // Proceedings of the 15th IFIP WG 6.2 International Conference on Wired/Wireless Internet Communications (WWIC, St. Petersburg, Russian Federation, 21–23 June 2017. Lecture Notes in Computer Science. Cham: Springer, 2017. Vol. 10372. PP. 26–37. DOI:10.1007/978-3-319-61382-6_3
4. Гетьман А.И., Иконникова М.К. Обзор методов классификации сетевого трафика с использованием машинного обучения // Труды института системного программирования РАН. 2020. Т. 32. № 6. С. 137–154. DOI:10.15514/ISPRAS-2020-32(6)-11
5. Черниговский А.В., Кривов М.В. Нейронные сети как инструмент анализа сетевого трафика // Вестник Ангарского государственного технического университета. 2019. № 13. С. 151–157. DOI:10.36629/2686-777x-2019-1-13-151-157
6. Гетьман А.И., Евстропов Е.Ф., Маркин Ю.В. Анализ сетевого трафика в режиме реального времени: обзор прикладных задач, подходов и решений // Препринт ИСП РАН. 2015. Т. 28. С. 1–52.
7. Ghosh A., Senthilrajan A. Classifying Network Traffic Using DPI And DFI // International Journal of Scientific & Technology Research. Lecture Notes on Data Engineering and Communications Technologies. 2019. Vol. 8. Iss. 11. PP. 3983–3988.
8. Процкая Е.П., Гай В.Е. Программная система анализа сетевого трафика // XXV Международная научно-техническая конференция «Информационные системы и технологии – 2019 (Нижний Новгород, Российская Федерация, 19 апреля 2019). Нижний Новгород: Нижегородский государственный технический университет им. Р.Е. Алексеева, 2019. С. 876–881.
9. Hu L., Zhang L. Real-time internet traffic identification based on decision tree // Proceedings of the World Automation Congress (Puerto Vallarta, Mexico, 24–28 June 2012). IEEE, 2012.
10. Deebalakshmi R., Jyothi V.L. A survey of classification algorithms for network traffic // Proceedings of the Second International Conference on Science Technology Engineering and Management (ICONSTEM, Chennai, India, 30–31 March 2016). IEEE, 2016. PP. 151–156. DOI:10.1109/ICONSTEM.2016.7560941
11. Karagiannis T., Broido A., Brownlee N., Claffy K.C., Faloutsos M. Is P2P dying or just hiding [P2P traffic measurement] // Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM, Dallas, USA, 29 November 2004 – 03 December 2004. IEEE, 2005. DOI:10.1109/GLOCOM.2004.1378239
12. Kohout J., Pevny T. Network Traffic Fingerprinting Based on Approximated Kernel Two-Sample Test // IEEE Transactions on Information Forensics and Security. 2018. Vol. 13. Iss. 3. PP. 788–801. DOI:10.1109/TIFS.2017.2768018
13. Perera P., Tian Y.C., Fidge C., Kelly W. A Comparison of Supervised Machine Learning Algorithms for Classification of Communications Network Traffic // Proceedings of the 24th International Conference on International Conference on Neural Information Processing (ICONIP, Guangzhou, China, 14–18 November 2017). Lecture Notes in Computer Science. Cham: Springer, 2017. Vol. 10634. PP. 445–454. DOI:10.1007/978-3-319-70087-8_47
14. Shi H., Li H., Zhang D., Cheng C., Wu W. Efficient and robust feature extraction and selection for traffic classification // Computer Networks. 2017. Vol. 119. PP. 1–16. DOI:10.1016/j.comnet.2017.03.011
15. Han J., Kamber M., Pie J. Data Mining: Concept and Techniques. Elsevier, 2006.
16. Kalyan G., Lakshmi A.J. Performance Assessment of Different Classification Techniques for Intrusion Detection // JORS Journal of Computer Engineering. 2012. Vol. 7. Iss. 5. PP. 2278–8727.
17. Protić D. Review of KDD CUP '99, NSL-KDD and Kyoto 2006+ datasets // Vojnotehnicki glasnik. 2018. Vol. 66. Iss. 3. PP. 580–596 DOI:10.5937/vojtehg66-16670
18. Lotfollahi M., Zade R.S.H., Siavoshani M.J., Saberian M. Deep packet: a novel approach for encrypted traffic classification using deeplearning // Soft Computing. 2020. Vol. 24. PP. 1999–2012. DOI:10.1007/s00500-019-04030-2
19. Катасёв А.С., Катасёва Д.В., Кирпичников А.П. Нейросетевая диагностика аномальной сетевой активности // Вестник технологического университета. 2015. Т. 18. № 6. С. 163–167.
20. Singh K., Agrawal S. Performance Analysis of Back Propagation Neural Network for Internet Traffic Classification // Proceedings of the National Conference on Recent Advances in Electronics and Communication Technologies (RAECT – 2011). 2011.
21. Manju N. Multilayer Feedforward Neural Network for Internet Traffic Classification // Special Issue on Soft Computing. 2023. DOI:10.9781/ijimai.2019.11.002

22. Абдурахманов Р.П., Тожиева Ф.К. Исследование систем управления трафиком на базе моделей нейронных сетей // Наука и мир. 2020. № 4-1; (80):26–32.
23. Ganowicz A., Starosta B., Knapieńska A., Walkowiak K. Short-Term Network Traffic Prediction with Multilayer Perceptron // Proceedings of the 6th SLAAI International Conference on Artificial Intelligence (SLAAI-ICAI, Colombo, Sri Lanka, 01–02 December 2022). IEEE, 2022. DOI:10.1109/SLAAI-ICAI56923.2022.10002431
24. Bikmukhamedov R.F., Nadeev A.F. Multi-Class Network Traffic Generators and Classifiers Based on Neural Networks // Systems of Signals Generating and Processing in the Field of on Board Communications (Moscow, Russian Federation, 16–18 March 2021). IEEE, 2021. DOI:10.1109/IEEECONF51389.2021.9416067
25. Azari A., Papapetrou P., Denic S., Peters G. Cellular Traffic Prediction and Classification: A Comparative Evaluation of LSTM and ARIMA // Proceedings of the 22nd International Conference on Discovery Science (DS, Split, Croatia, 28–30 October 2019). Lecture Notes in Computer Science. Cham: Springer, 2019. Vol. 11828. PP. 129–144. DOI:10.1007/978-3-030-33778-0_11
26. Yang L., Wang Z., Feng Y., Yan H. An Effective Real-time Traffic Classification Method Using Convolutional Neural Network // Research Square. 2023. DOI:10.21203/rs.3.rs-3224251/v1
27. Chen X., Wang P., Yu J. CNN based encrypted traffic identification method // Journal of Nanjing University of Posts and Telecommunications (Natural Science). 2018. Vol. 38. PP. 36–41. DOI:10.14132/j.cnki.1673-5439.2018.06.006
28. Guo L., Wu Q., Liu S., Duan M., Li H., Sun J. Deep learning-based real-time VPN encrypted traffic identification methods // Journal of Real-Time Image Processing. 2020. Vol. 17. PP. 103–114. DOI:10.1007/s11554-019-00930-6
29. Yang J., Narantuya J., Lim H. Bayesian Neural Network based Encrypted Traffic Classification using Initial Handshake Packets // Proceedings of the 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks – Supplemental Volume (DSN-S, Portland, USA, 24–27 June 2019). IEEE, 2019. DOI:10.1109/DSN-S.2019.00015
30. Izadi S., Ahmadi M., Nikbazzm R. Analysis of Feature Selection Methods for Network Traffic Classification // Proceedings of the 8th International Conference on Advanced Intelligent Systems and Informatics (AISI, Cairo, Egypt, 20–22 November 2022). Lecture Notes on Data Engineering and Communications Technologies. Cham: Springer, 2023. Vol. 152. PP. 65–77. DOI:10.1007/978-3-031-20601-6_6
31. Yamansavascilar B., Guvensan M.A., Yavuz A.G., Karsligil M.E. Application identification via network traffic classification // Proceedings of the International Conference on Computing, Networking and Communications (ICNC, Silicon Valley, USA, 26–29 January 2017). IEEE, 2017. DOI:10.1109/ICNC.2017.7876241
32. Kwon J., Lee J., Yu M., Park H. Automatic Classification of Network Traffic Data based on Deep Learning in ONOS Platform // Proceedings of the International Conference on Information and Communication Technology Convergence (ICTC, Jeju, Korea (South), 21–23 October 2020). IEEE, 2020. DOI:10.1109/ICTC49870.2020.9289257
33. Wang W., Zeng X., Jinlin W. End-to-end encrypted traffic classification with one-dimensional convolution neural networks // Proceedings of the International Conference on Intelligence and Security Informatics (ISI, Beijing, China, 22–24 July 2017). IEEE, 2018. DOI:10.1109/ISI.2017.8004872
34. Tooke J., Chavula J. Resource-Constrained Real-Time Network Traffic Classification Using One-Dimensional Convolutional Neural Networks // Proceedings of the 13th EAI International Conference on e-Infrastructure and e-Services for Developing Countries (AFRICOMM, Zanzibar, Tanzania, 1–3 December 2021). Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Cham: Springer, 2022. Vol. 443. PP. 107–127. DOI:10.1007/978-3-031-06374-9_8
35. Izadi S., Ahmadi M., Nikbazzm R. Network traffic classification using convolutional neural network and ant-lion optimization // Computers & Electrical Engineering. 2022. Vol. 101. P. 108024. DOI:10.1016/j.compeleceng.2022.108024
36. Wijesekara P.A.D.S.N., Gunawardena S.A. Comprehensive Survey on Knowledge-Defined Networking // Telecom. 2023. Vol. 4. Iss. 43. PP. 477–596. DOI:10.3390/telecom4030025
37. Jarvis M.P., Nuzzo-Jones G., Heffernan N.T. Applying Machine Learning Techniques to Rule Generation in Intelligent Tutoring Systems // Proceedings of the 7th International Conference on Intelligent Tutoring Systems (ITS, Maceió, Brazil, 30 August – 3 September 2004). Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2004. Vol. 3220. PP. 541–553. DOI:10.1007/978-3-540-30139-4_51
38. Boley H., Tabet S., Wagner G. Design Rationale for RuleML: A Markup Language for Semantic Web Rules // Proceedings of the first Semantic Web Working Symposium (SWWS, Stanford, USA, 30 July – 1 August 2001). Stanford University, 2001. Vol. 1. PP. 381–401.
39. Kifer M. Rule Interchange Format: The Framework // Proceedings of the Second International Conference on Web Reasoning and Rule Systems (RR, Karlsruhe, Germany, 31 October – 1 November 2008). Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2008. Vol. 5341. PP. 1–11. DOI:10.1007/978-3-540-88737-9_1
40. Horrocks I., Patel-Schneider P.F., Boley H., Tabet S., Grosof B., Dean M. SWRL: A Semantic Web Rule Language Combining OWL and RuleML // W3C Member Submission. 2004. PP. 1–31.
41. Wu D., Li Z., Wang J., Zheng Y., Li M., Huang Q. Vision and Challenges for Knowledge Centric Networking // IEEE Wireless Communications. 2019. Vol. 26. Iss. 4. PP. 117–123. DOI:10.1109/MWC.2019.1800323
42. Narisetty R., Dane L., Malishevskiy A., Gurkan D., Bailey S., Narayan S., et al. OpenFlow Configuration Protocol: Implementation for the of Management Plane // Proceedings of the Second GENI Research and Educational Experiment Workshop (Salt Lake City, USA, 20–22 March 2013). IEEE, 2003. PP. 66–67. DOI:10.1109/GREE.2013.21
43. Safrianti E., Sari L.O., Sari N.A. Real-Time Network Device Monitoring System with Simple Network Management Protocol (SNMP) Model // Proceedings of the 3rd International Conference on Research and Academic Community Services (ICRACOS, Surabaya, Indonesia, 9–10 October 2021). IEEE, 2021. PP. 122–127. DOI:10.1109/ICRACOS53680.2021.9701973
44. Wijesekara P.A.D.S.N., Sudheera K.L.K., Sandamali G.G.N., Chong P.H.J. An Optimization Framework for Data Collection in Software Defined Vehicular Networks // Sensors. 2023. Vol. 23. Iss. 3. P. 1600. DOI:10.3390/s23031600

45. Wette P., Karl H. Which flows are hiding behind my wildcard rule? // Proceedings of the conference on SIGCOMM (Hong Kong, China, 12–16 August 2013). New York: ACM, 2013. PP. 541–542. DOI:10.1145/2486001.2491710
46. Zhou D., Yan Z., Liu G. Atiquzzaman, M. An Adaptive Network Data Collection System in SDN // IEEE Transactions on Cognitive Communications and Networking. 2020. Vol. 6. Iss. 2. PP. 562–574. DOI:10.1109/TCCN.2019.2956141
47. Liao W.H., Kuai S.C. An Energy-Efficient SDN-Based Data Collection Strategy for Wireless Sensor Networks // Proceedings of the 7th International Symposium on Cloud and Service Computing (SC2, Kanazawa, Japan, 22–25 November 2017). IEEE, 2017. PP. 91–97. DOI:10.1109/SC2.2017.21
48. Bjorklund M. YANG – A Data Modeling Language for the Network Configuration Protocol (NETCONF). URL: <https://www.rfc-editor.org/rfc/rfc6020> (Accessed 19.10.2023)
49. Uslar M., Specht M., Rohjans S., Trefke J., González J.M. The Common Information Model CIM: IEC 61968/61970 and 62325 – A Practical Introduction to the CIM. Berlin, Heidelberg: Springer, 2012. DOI:10.1007/978-3-642-25215-0
50. Gude N., Koponen, T., Pettit J., Pfaff B., Casado M., McKeown N., Shenker S. NOX: towards an operating system for networks // ACM SIGCOMM Computer Communication Review. 2008. Vol. 38. Iss. 3. PP. 105–110. DOI:10.1145/1384609.1384625
51. Rowshanrad S., Abdi V., Keshtgari M. Performance evaluation of SDN controllers: Floodlight and OpenDaylight // IIUM Engineering Journal. 2016. Vol. 17. Iss. 2. PP. 47–57. DOI:10.31436/iiumej.v17i2.615
52. Sanvito D., Moro D., Gulli M., Filippini I., Capone A., Campanella A. ONOS Intent Monitor and Reroute service: Enabling plug&play routing logic // Proceedings of the 4th Conference on Network Softwarization and Workshops (NetSoft, Montreal, Canada, 25–29 June 2018). IEEE, 2018. PP. 272–276. DOI:10.1109/NETSOFT.2018.8460064
53. Дмитриева Ю.С. Управление сетевыми ресурсами на основе намерений // Вестник связи. 2022. № 4. С. 20–26.
54. Rotsos C., King D., Farshad A., Bird J., Fawcett L., Georgalas N., et al. Network service orchestration standardization: A technology survey // Computer Standards & Interfaces. 2017. Vol. 54. Part 4. PP. 203–215. DOI:10.1016/j.csi.2016.12.006
55. Bannour F., Souihi S., Mellouk A. Distributed SDN Control: Survey, Taxonomy, and Challenges // IEEE Communications Surveys & Tutorials. 2017. Vol. 20. Iss. 1. PP. 333–354. DOI:10.1109/COMST.2017.2782482
56. Sanvito D., Moro D., Gulli M., Filippini I., Capone A., Campanella A. ONOS Intent Monitor and Reroute service: Enabling plug&play routing logic // Proceedings of the 4th Conference on Network Softwarization and Workshops (NetSoft, Montreal, Canada, 25–29 June 2018). IEEE, 2018. PP. 272–276. DOI:10.1109/NETSOFT.2018.8460064
57. Koponen T., Casado M., Gude N., Stribling J., Poutievski L., Zhu M., et al. Onix: A Distributed Control Platform for Large-Scale Production Networks // Proceedings of the 9th USENIX Conference on Operating Systems Design and Implementation (OSDI, Vancouver, Canada, 4–6 October 2010). Berkeley: USENIX Association, 2010. PP. 351–364.
58. Zhu M., Cao J., Pang D., He Z., Xu M. SDN-Based Routing for Efficient Message Propagation in VANET // Proceedings of the 10th International Conference on Wireless Algorithms, Systems, and Applications (WASA, Qufu, China, 10–12 August 2015). Lecture Notes in Computer Science. Cham: Springer, 2015. Vol. 9204. PP. 788–797. DOI:10.1007/978-3-319-21837-3_77
59. Moghaddam F.F., Wieder P., Yahyapour R. Policy Engine as a Service (PEaaS): An approach to a Reliable Policy Management Framework in Cloud Computing Environments // Proceedings of the 4th International Conference on Future Internet of Things and Cloud (FiCloud, Vienna, Austria, 22–24 August 2016). IEEE, 2016. PP. 137–144. DOI:10.1109/FiCloud.2016.27
60. Chen Y.J., Wang L.C., Lin F.Y., Lin B.S.P. Deterministic Quality of Service Guarantee for Dynamic Service Chaining in Software Defined Networking // IEEE Transactions on Network and Service Management. 2017. Vol. 14. Iss. 4. PP. 991–1002. DOI:10.1109/TNSM.2017.2758328
61. Yang G., Jin H., Kang M., Moon G.J., Yoo C. Network Monitoring for SDN Virtual Networks // Proceedings of the Conference on Computer Communications (IEEE INFOCOM, Toronto, Canada, 06–09 July 2020). 2020. PP. 1261–1270. DOI:10.1109/INFOCOM41043.2020.9155260
62. Ahvar E., Ahvar S., Raza S.M., Vilchez J. M.S., Lee G.M. Next Generation of SDN in Cloud-Fog for 5G and Beyond-Enabled Applications: Opportunities and Challenges // Network. 2021. Vol. 1. Iss. 1. PP. 28–49. DOI:10.3390/network1010004
63. Voellmy A., Kim H., Feamster N. Protera: a language for high-level reactive network control // Proceedings of the First Workshop on Hot Topics in Software Defined Networks (HotSDN, Helsinki, Finland, 13 August 2012). New York: ACM, 2012. PP. 43–48. DOI:10.1145/2342441.2342451
64. Voellmy A., Hudak P. Nettle: Functional Reactive Programming for OpenFlow Networks. URL: <https://pages.cs.wisc.edu/~akella/CS838/F12/838-CloudPapers/Nettle.pdf> (Accessed 20.12.2023)
65. Foster N., Freedman M.J., Harrison R., Rexford J., Meola M.L., Walker D. Frenetic: a high-level language for OpenFlow networks // Proceedings of the Workshop on Programmable Routers for Extensible Services of Tomorrow (PRESTO, Philadelphia, USA, 30 November 2010). New York: ACM, 2010. PP. 1–6. DOI:10.1145/1921151.1921160
66. Kim H., Reich J., Gupta A., Shahbaz M., Feamster N., Clark R. Kinetic: Verifiable Dynamic Network Control // Proceedings of the 12th USENIX Symposium on Networked Systems Design and Implementation (NSDI, Oakland, USA, 4–6 May 2015). Berkeley: USENIX Association, 2015. PP. 59–72.

References

1. Dmitrieva J. Comparative Analysis of Network Resource Management Methods in SDN. *Proceedings of Telecommun. Univ.* 2022;8(1):73–83. DOI:10.31854/1813-324X-2022-8-1-73-83
2. Kirichek R., Vladiko A., Zakharov M., Koucheryavy A. Model networks for Internet of Things and SDN. *Proceedings of the 18th International Conference on Advanced Communication Technology, ICACT, 31 January 2016 – 03 February 2016, PyeongChang, Korea (South)*. IEEE; 2016. p.76–79. DOI:10.1109/ICAICT.2016.7423280
3. Muhizi S., Shamshin G., Muthanna A., Kirichek R., Vladiko A., Koucheryavy A. Analysis and Performance Evaluation of SDN Queue Model. *Proceedings of the 15th IFIP WG 6.2 International Conference on Wired/Wireless Internet Communications*,

- WWIC, 21–23 June 2017, St. Petersburg, Russian Federation. *Lecture Notes in Computer Science*, vol.10372. Cham: Springer; 2017. p.26–37. DOI:10.1007/978-3-319-61382-6_3
4. Getman A.I., Ikonnikov M.K. A survey of network traffic classification methods using machine learning. *Proceedings of ISP RAS*. 2020;32(6):137–154. DOI:10.15514/ISPRAS-2020-32(6)-11
5. Chernigovskiy A.V., Krivov M.V. Neural networks as an instrument of analysis of network traffic. *Bulletin of the Angarsk State Technical University*. 2019;13:151–157. DOI:10.36629/2686-777x-2019-1-13-151-157
6. Getman A.I., Evstropov E.F., Markin Y.V. Wirespeed network traffic analysis: survey of applied problems, approaches and solutions. *ISP RAS preprints*. 2015;28:1–52.
7. Ghosh A., Senthilrajan A. Classifying Network Traffic Using DPI And DFI. *International Journal of Scientific & Technology Research*. 2019;8(11):3983–3988.
8. Prockaya E.P., Gai V.E. Software system analysis of network traffic. *Proceedings of the XXV International Scientific and Technical Conference on Information Systems and Technologies – 2019, 19 April 2019, Nizhny Novgorod, Russian Federation*. Nizhny Novgorod: Nizhny Novgorod State Technical University Publ.; 2019. p.876–881.
9. Hu L., Zhang L. Real-time internet traffic identification based on decision tree. *Proceedings of the World Automation Congress, 24–28 June 2012, Puerto Vallarta, Mexico*. IEEE; 2012.
10. Deebalakshmi R., Jyothi V.L. A survey of classification algorithms for network traffic. *Proceedings of the Second International Conference on Science Technology Engineering and Management, ICONSTEM, 30–31 March 2016, Chennai, India*. IEEE; 2016. p.151–156. DOI:10.1109/ICONSTEM.2016.7560941
11. Karagiannis T., Broido A., Brownlee N., Claffy K.C., Faloutsos M. Is P2P dying or just hiding [P2P traffic measurement]. *Proceedings of the IEEE Global Telecommunications Conference, GLOBECOM, 29 November 2004 – 03 December 2004, Dallas, USA*. IEEE; 2005. DOI:10.1109/GLOCOM.2004.1378239
12. Kohout J., Pevny T. Network Traffic Fingerprinting Based on Approximated Kernel Two-Sample Test. *IEEE Transactions on Information Forensics and Security*. 2018;13(3). DOI:10.1109/TIFS.2017.2768018
13. Perera P., Tian Y.C., Fidge C., Kelly W. A Comparison of Supervised Machine Learning Algorithms for Classification of Communications Network Traffic. *Proceedings of the 24th International Conference on International Conference on Neural Information Processing, ICONIP, 14–18 November 2017, Guangzhou, China. Lecture Notes in Computer Science, vol.10634*. Cham: Springer; 2017. p. 445–454. DOI:10.1007/978-3-319-70087-8_47
14. Shi H., Li H., Zhang D., Cheng C., Wu W. Efficient and robust feature extraction and selection for traffic classification. *Computer Networks*. 2017;119:1–16. DOI:10.1016/j.comnet.2017.03.011
15. Han J., Kamber M., Pie J. *Data Mining: Concept and Techniques*. Elsevier: 2006.
16. Kalyan G., Lakshmi A.J. Performance Assessment of Different Classification Techniques for Intrusion Detection. *JORS Journal of Computer Engineering*. 2012;7(5):2278–8727.
17. Protic D. Review of KDD CUP '99, NSL-KDD and Kyoto 2006+ datasets. *Vojnotehnicki glasnik*. 2018;66(3):580–596 DOI:10.5937/vojtehg66-16670
18. Lotfollahi M., Zade R.S.H., Siavoshani M.J., Saberian M. Deep packet: a novel approach for encrypted traffic classification using deeplearning. *Soft Computing*. 2020;24:1999–2012. DOI:10.1007/s00500-019-04030-2
19. Katasev A.S., Kataseva D.V., Kirpichnikov A.P. Neural network diagnosis of abnormal network activity. *Herald of Tekhnological Universite*. 2015;18(6):163–167.
20. Singh K., Agrawal S. Performance Analysis of Back Propagation Neural Network for Internet Traffic Classification. *Proceedings of the National Conference on Recent Advances in Electronics and Communication Technologies, RAECT – 2011*. 2011
21. Manju N. Multilayer Feedforward Neural Network for Internet Traffic Classification. *Special Issue on Soft Computing*. 2023. DOI:10.9781/ijimai.2019.11.002
22. Abdurakhmanov R.P., Tojjeva F.Q. The research of traffic management systems based on neural network models. *Science and world*. 2020;4-1(80):26–32.
23. Ganowicz A., Starosta B., Knapińska A., Walkowiak K. Short-Term Network Traffic Prediction with Multilayer Perceptron. *Proceedings of the 6th SLAAI International Conference on Artificial Intelligence, SLAAI-ICAI, 01–02 December 2022, Colombo, Sri Lanka*. IEEE; 2022. DOI:10.1109/SLAAI-ICAI56923.2022.10002431
24. Bikmukhamedov R.F., Nadeev A.F. Multi-Class Network Traffic Generators and Classifiers Based on Neural Networks. *Systems of Signals Generating and Processing in the Field of on Board Communications, 16–18 March 2021, Moscow, Russian Federation*. IEEE; 2021. DOI:10.1109/IEEECONF51389.2021.9416067
25. Azari A., Papapetrou P., Denic S., Peters G. Cellular Traffic Prediction and Classification: A Comparative Evaluation of LSTM and ARIMA. *Proceedings of the 22nd International Conference on Discovery Science, DS, 28–30 October 2019, Split, Croatia. Lecture Notes in Computer Science, vol.11828*. Cham: Springer; 2019. p.129–144. DOI:10.1007/978-3-030-33778-0_11
26. Yang L., Wang Z., Feng Y., Yan H. An Effective Real-time Traffic Classification Method Using Convolutional Neural Network. *Research Square*. 2023. DOI:10.21203/rs.3.rs-3224251/v1
27. Chen X., Wang P., Yu J. CNN based encrypted traffic identification method. *Journal of Nanjing University of Posts and Telecommunications (Natural Science)*. 2018;38:36–41. DOI:10.14132/j.cnki.1673-5439.2018.06.006
28. Guo L., Wu Q., Liu S., Duan M., Li H., Sun J. Deep learning-based real-time VPN encrypted trafrc identification methods. *Journal of Real-Time Image Processing*. 2020;17:103–114. DOI:10.1007/s11554-019-00930-6
29. Yang J., Narantuya J., Lim H. Bayesian Neural Network based Encrypted Traffic Classification using Initial Handshake Packets. *Proceedings of the 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks – Supplemental Volume, DSN-S, 24–27 June 2019, Portland, USA*. IEEE; 2019. DOI:10.1109/DSN-S.2019.00015
30. Izadi S., Ahmadi M., Nikbazm R. Analysis of Feature Selection Methods for Network Traffic Classification. *Proceedings of the 8th International Conference on on Advanced Intelligent Systems and Informatics, AISI, 20–22 November 2022, Cairo, Egypt*.

31. Yamansavascular B., Guvensan M.A., Yavuz A.G., Karsligil M.E. Application identification via network traffic classification. *Proceedings of the International Conference on Computing, Networking and Communications, ICNC, 26–29 January 2017, Silicon Valley, USA*. IEEE; 2017. DOI:10.1109/ICNC.2017.7876241
32. Kwon J., Lee J., Yu M., Park H. Automatic Classification of Network Traffic Data based on Deep Learning in ONOS Platform. *Proceedings of the International Conference on Information and Communication Technology Convergence, ICTC, 21–23 October 2020, Jeju, Korea (South)*. IEEE; 2020. DOI:10.1109/ICTC49870.2020.9289257
33. Wang W., Zeng X., Jinlin W. End-to-end encrypted traffic classification with one-dimensional convolution neural networks. *Proceedings of the International Conference on Intelligence and Security Informatics, ISI, 22–24 July 2017, Beijing, China*. IEEE; 2018. DOI:10.1109/ISI.2017.8004872
34. Tooke J., Chavula J. Resource-Constrained Real-Time Network Traffic Classification Using One-Dimensional Convolutional Neural Networks. *Proceedings of the 13th EAI International Conference on e-Infrastructure and e-Services for Developing Countries, AFRICOMM, 1–3 December 2021, Zanzibar, Tanzania. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol.443*. Cham: Springer; 2022. p.107–127. DOI:10.1007/978-3-031-06374-9_8
35. Izadi S., Ahmadi M., Nikbazzam R. Network traffic classification using convolutional neural network and ant-lion optimization. *Computers & Electrical Engineering*. 2022;101:108024. DOI:10.1016/j.compeleceng.2022.108024
36. Wijesekara P.A.D.S.N., Gunawardena S.A. Comprehensive Survey on Knowledge-Defined Networking. *Telecom*. 2023; 4(43):477–596. DOI:10.3390/telecom4030025
37. Jarvis M.P., Nuzzo-Jones G., Heffernan N.T. Applying Machine Learning Techniques to Rule Generation in Intelligent Tutoring Systems. *Proceedings of the 7th International Conference on Intelligent Tutoring Systems, ITS, 30 August – 3 September 2004, Maceio, Brazil. Lecture Notes in Computer Science, vol.3220*. Berlin, Heidelberg: Springer; 2004. p.541–553. DOI:10.1007/978-3-540-30139-4_51
38. Boley H., Tabet S., Wagner G. Design Rationale for RuleML: A Markup Language for Semantic Web Rules. *Proceedings of the first Semantic Web Working Symposium, SWWS, 30 July – 1 August 2001, Stanford, USA, vol.1*. Stanford University; 2001. p.381–401.
39. Kifer M. Rule Interchange Format: The Framework. *Proceedings of the Second International Conference on Web Reasoning and Rule Systems, RR, 31 October – 1 November 2008, Karlsruhe, Germany. Lecture Notes in Computer Science, vol.5341*. Berlin, Heidelberg: Springer; 2008. p.1–11. DOI:10.1007/978-3-540-88737-9_1
40. Horrocks I., Patel-Schneider P.F., Boley H., Tabet S., Grosz B., Dean M. SWRL: A Semantic Web Rule Language Combining OWL and RuleML. *W3C Member Submission*. 2004:1–31.
41. Wu D., Li Z., Wang J., Zheng Y., Li M., Huang Q. Vision and Challenges for Knowledge Centric Networking. *IEEE Wireless Communications*. 2019;26(4):117–123. DOI:10.1109/MWC.2019.1800323
42. Narisetty R., Dane L., Malishevskiy A., Gurkan D., Bailey S., Narayan S., et al. OpenFlow Configuration Protocol: Implementation for the of Management Plane. *Proceedings of the Second GENI Research and Educational Experiment Workshop, 20–22 March 2013, Salt Lake City, USA*. IEEE; 2003. p.66–67. DOI:10.1109/GREE.2013.21
43. Safrianti E., Sari L.O., Sari N.A. Real-Time Network Device Monitoring System with Simple Network Management Protocol (SNMP) Model. *Proceedings of the 3rd International Conference on Research and Academic Community Services, ICRACOS, 9–10 October 2021, Surabaya, Indonesia*. IEEE; 2021. p.122–127. DOI:10.1109/ICRACOS53680.2021.9701973
44. Wijesekara P.A.D.S.N., Sudheera K.L.K., Sandamali G.G.N., Chong P.H.J. An Optimization Framework for Data Collection in Software Defined Vehicular Networks. *Sensors*. 2023;23(3):1600. DOI:10.3390/s23031600
45. Wette P., Karl H. Which flows are hiding behind my wildcard rule? *Proceedings of the conference on SIGCOMM, 12–16 August 2013, Hong Kong, China*. New York: ACM; 2013. p.541–542. DOI:10.1145/2486001.2491710
46. Zhou D., Yan Z., Liu G. Atiquzzaman, M. An Adaptive Network Data Collection System in SDN. *IEEE Transactions on Cognitive Communications and Networking*. 2020;6(2):562–574. DOI:10.1109/TCCN.2019.2956141
47. Liao W.H., Kuai S.C. An Energy-Efficient SDN-Based Data Collection Strategy for Wireless Sensor Networks. *Proceedings of the 7th International Symposium on Cloud and Service Computing, SC2, 22–25 November 2017, Kanazawa, Japan*. IEEE; 2017. p.91–97. DOI:10.1109/SC2.2017.21
48. Bjorklund M. YANG – A Data Modeling Language for the Network Configuration Protocol (NETCONF). URL: <https://www.rfc-editor.org/rfc/rfc6020> [Accessed 19.10.2023]
49. Uslar M., Specht M., Rohjans S., Trefke J., González J.M. *The Common Information Model CIM: IEC 61968/61970 and 62325 – A Practical Introduction to the CIM*. Berlin, Heidelberg: Springer; 2012. DOI:10.1007/978-3-642-25215-0
50. Gude N., Koponen, T., Pettit J., Pfaff B., Casado M., McKeown N., Shenker S. NOX: towards an operating system for networks. *ACM SIGCOMM Computer Communication Review*. 2008;38(3):105–110. DOI:10.1145/1384609.1384625
51. Rowshanrad S., Abdi V., Keshtgari M. Performance evaluation of SDN controllers: Floodlight and OpenDaylight. *IIUM Engineering Journal*. 2016;17(2):47–57. DOI:10.31436/iiumej.v17i2.615
52. Sanvito D., Moro D., Gulli M., Filippini I., Capone A., Campanella A. ONOS Intent Monitor and Reroute service: Enabling plug&play routing logic. *Proceedings of the 4th Conference on Network Softwarization and Workshops, NetSoft, 25–29 June 2018, Montreal, Canada*. IEEE; 2018. p.272–276. DOI:10.1109/NETSOFT.2018.8460064
53. Dmitrieva J. Intent-Based Networking Management. *Communication Bulletin*. 2022;4:20–26.
54. Rotsos C., King D., Farshad A., Bird J., Fawcett L., Georgalas N., et al. Network service orchestration standardization: A technology survey. *Computer Standards & Interfaces*. 2017;54:203–215. DOI:10.1016/j.csi.2016.12.006
55. Bannour F., Souihi S., Mellouk A. Distributed SDN Control: Survey, Taxonomy, and Challenges. *IEEE Communications Surveys & Tutorials*. 2017;20(1):333–354. DOI:10.1109/COMST.2017.2782482
56. Sanvito D., Moro D., Gulli M., Filippini I., Capone A., Campanella A. ONOS Intent Monitor and Reroute service: Enabling plug&play routing logic. *Proceedings of the 4th Conference on Network Softwarization and Workshops, NetSoft, 25–29 June 2018, Montreal, Canada*. IEEE; 2018. p.272–276. DOI:10.1109/NETSOFT.2018.8460064

57. Koponen T., Casado M., Gude N., Stribling J., Poutievski L., Zhu M., et al. Onix: A Distributed Control Platform for Large-Scale Production Networks. *Proceedings of the 9th USENIX Conference on Operating Systems Design and Implementation, OSDI, 4–6 October 2010, Vancouver, Canada*. Berkeley: USENIX Association; 2010. p.351–364.
58. Zhu M., Cao J., Pang D., He Z., Xu M. SDN-Based Routing for Efficient Message Propagation in VANET. *Proceedings of the 10th International Conference on Wireless Algorithms, Systems, and Applications, WASA, 10–12 August 2015, Qufu, China. Lecture Notes in Computer Science, vol.9204*. Cham: Springer; 2015. p.788–797. DOI:10.1007/978-3-319-21837-3_77
59. Moghaddam F.F., Wieder P., Yahyapour R. Policy Engine as a Service (PEaaS): An approach to a Reliable Policy Management Framework in Cloud Computing Environments. *Proceedings of the 4th International Conference on Future Internet of Things and Cloud, FiCloud, 22–24 August 2016, Vienna, Austria*. IEEE; 2016. p.137–144. DOI:10.1109/FiCloud.2016.27
60. Chen Y.J., Wang L.C., Lin F.Y., Lin B.S.P. Deterministic Quality of Service Guarantee for Dynamic Service Chaining in Software Defined Networking. *IEEE Transactions on Network and Service Management*. 2017;14(4):991–1002. DOI:10.1109/TNSM.2017.2758328
61. Yang G., Jin H., Kang M., Moon G.J., Yoo C. Network Monitoring for SDN Virtual Networks. *Proceedings of the Conference on Computer Communications, IEEE INFOCOM, 06–09 July 2020, Toronto, Canada*. IEEE; 2020. p.1261–1270. DOI:10.1109/INFOCOM41043.2020.9155260
62. Ahvar E., Ahvar S., Raza S.M., Vilchez J. M.S., Lee G.M. Next Generation of SDN in Cloud-Fog for 5G and Beyond-Enabled Applications: Opportunities and Challenges. *Network*. 2021;1(1):28–49. DOI:10.3390/network1010004
63. Voellmy A., Kim H., Feamster N. Proccera: a language for high-level reactive network control. *Proceedings of the First Workshop on Hot Topics in Software Defined Networks, HotSDN, 13 August 2012, Helsinki, Finland*. New York: ACM; 2012. p.43–48. DOI:10.1145/2342441.2342451
64. Voellmy A., Hudak P. *Nettle: Functional Reactive Programming for Openflow Networks*. URL: <https://pages.cs.wisc.edu/~akella/CS838/F12/838-CloudPapers/Nettle.pdf> [Accessed 20.12.2023]
65. Foster N., Freedman M.J., Harrison R., Rexford J., Meola M.L., Walker D. Frenetic: a high-level language for OpenFlow networks. *Proceedings of the Workshop on Programmable Routers for Extensible Services of Tomorrow, PRESTO, 30 November 2010, Philadelphia, USA*. New York: ACM; 2010. p.1–6. DOI:10.1145/1921151.1921160
66. Kim H., Reich J., Gupta A., Shahbaz M., Feamster N., Clark R. Kinetic: Verifiable Dynamic Network Control. *Proceedings of the 12th USENIX Symposium on Networked Systems Design and Implementation, NSDI, 4–6 May 2015, Oakland, USA*. Berkeley: USENIX Association; 2015. p.59–72.

Статья поступила в редакцию 01.11.2023; одобрена после рецензирования 24.11.2023; принята к публикации 15.12.2023.

The article was submitted 01.11.2023; approved after reviewing 24.11.2023; accepted for publication 15.12.2023.

Информация об авторах:

ДМИТРИЕВА
Юлия Сергеевна

ассистент кафедры инфокоммуникационных сетей и систем Санкт-Петербургского государственного университета телекоммуникаций им. проф. М.А. Бонч-Бруевича
 <https://orcid.org/0000-0002-7736-7121>

ОКУНЕВА
Дарина Владимировна

кандидат технических наук, декан факультета инфокоммуникационных сетей и систем Санкт-Петербургского государственного университета телекоммуникаций им. проф. М.А. Бонч-Бруевича
 <https://orcid.org/0009-0005-4241-8784>

ЕЛАГИН
Василий Сергеевич

кандидат технических наук, доцент, доцент кафедры инфокоммуникационных систем Санкт-Петербургского государственного университета телекоммуникаций им. проф. М.А. Бонч-Бруевича
 <https://orcid.org/0000-0003-4077-6869>

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ТЕЛЕКОММУНИКАЦИИ

**2.3.1 – Системный анализ,
управление и обработка
информации**

**2.3.6 – Методы и системы защиты
информации, информационная
безопасность**

Научная статья

УДК 004.056.5

DOI:10.31854/1813-324X-2023-9-6-59-67



Предложения по построению универсального фаззера протоколов

Дмитрий Александрович Васинев ✉, vda33@academ.msk.rsnet.ru

Михаил Викторович Соловьев, saintsdertr@gmail.com

Академия Федеральной службы охраны Российской Федерации,
Орел, 302015, Российская Федерация

Аннотация: В статье исследуется проблема обеспечения информационной безопасности в сфере телекоммуникаций с использованием метода фаззинга. Проведен анализ современных программных продуктов, предназначенных для тестирования методом фаззинга, выявлены их недостатки и предложен подход к созданию универсального мутационного фаззера протоколов. Способ формирования тестовых данных позволяет автоматизировать процесс поиска уязвимостей в телекоммуникационных протоколах и программном обеспечении. Его новизна заключается в формировании тестовых конструкций на основе параметров полей телекоммуникационных протоколов. Предложенное решение фаззера позволяет формировать вектор атаки на основе известных параметров, представленных в банке данных угроз, так и модифицировать эти вектора атак.

Ключевые слова: информационная безопасность, фаззинг, уязвимости, телекоммуникационные протоколы, генетический фаззер

Ссылка для цитирования: Васинев Д.А., Соловьев М.В. Предложения по построению универсального фаззера протоколов // Труды учебных заведений связи. 2023. Т. 9. № 6. С. 59–67. DOI:10.31854/1813-324X-2023-9-6-59-67

Proposals for Universal Protocol Fuzzer Construction

Dmitriy Vasinev ✉, vda33@academ.msk.rsnet.ru

Mikhail Solovev, saintsdertr@gmail.com

The Academy of the Federal Guard Service of Russian Federation,
Orel, 302015, Russian Federation

Abstract: The article studies the problem of ensuring information security in the field of telecommunications using the phasing method. The analysis of modern software products designed for testing by the fuzzing method is carried out, their disadvantages are revealed, and an approach to the creation of a universal mutation protocol fuzzer is proposed. The method of test data formation allows to automate the process of searching for vulnerabilities in telecommunication protocols and pro-software. Its novelty lies in the formation of test constructs on the basis of parameters of fields of telecommunication protocols. The proposed fuzzy solution allows to form an attack vector on the basis of known parameters presented in the threat database, as well as to modify these attack vectors.

Keywords: information security, fuzzing, vulnerabilities, telecommunication protocols, genetic fuzzing

For citation: Vasinev D., Solovev M. Proposals for Universal Protocol Fuzzer Construction. *Proceedings of Telecommunication Universities*. 2023;9(6):59–67. DOI:10.31854/1813-324X-2023-9-6-59-67

Введение

В связи с постоянным появлением новых угроз и уязвимостей значительные усилия специалистов по информационной безопасности посвящены разработке методов поиска уязвимостей в телекоммуникационных протоколах, программном обеспечении и информационных системах, функционирующих в объектах критической информационной инфраструктуры (КИИ). Предпосылкой к появлению уязвимостей является значительная размерность пространства состояний параметров протокольных конструкций, участвующих в обработке и передаче данных. Таким образом, актуальной задачей является автоматизация процесса поиска уязвимостей в протокольных конструкциях, разработка автоматических и автоматизированных методов, уменьшающих рутинные функции специалиста по информационной безопасности, связанные с поиском уязвимостей в телекоммуникационном оборудовании объектов КИИ.

Анализ существующих решений

В современных исследованиях в области тестирования программных средств актуальным направлением является фаззинг [1–3]. Под термином «фаззинг» (эквив. на русск. – тестирование) понимается методология тестирования, традиционно применяемая для исследования программ, в которых в качестве данных для анализа защищенности объекта применяются значения, выходящие за рамки корректных [1]. Под фаззером понимается специализированное программное обеспечение (ПО), на основе которого из исходных данных выполняется формирование тестируемой последовательности, осуществляется ее передача для обработки тестируемым объектам с целью обнаружения отклонений от нормального функционирования. Традиционным стало применение методологии фаззинга для тестирования безопасности программных средств.

Таким образом, исходя из работ [1, 2], наиболее близкой по функциональным возможностям поиска уязвимости в протокольных конструкциях является тестирование программ методом фаззинга. Известны смешанные программные и протокольные методы решения этой задачи. В рамках исследования разрабатываются функциональные и технические элементы, позволяющие осуществлять поиск уязвимостей в протокольных стеках объектов КИИ.

Основным отличием предлагаемого решения является тестирование не отдельного протокола, а группы протоколов, с учетом существующих служебных и сигнальных каналов передачи данных. Для ее решения необходимо провести анализ принципов построения таких анализаторов (рисунок 1), существующих функциональных возможностей (таблица 1), выделить направления их совершен-

ствования, сформировать предложения по совершенствованию методов тестирования протокольных стеков, осуществить верификацию предлагаемых решений.

Представленная на рисунке 1 классификация видов фаззинга позволяет выделить функциональные особенности, характерные как для программных, так и протокольных фаззеров. Для тестирования функциональных возможностей протокольных конструкций используются методы динамического изменения тестовых последовательностей – мутационные и генетические алгоритмы.

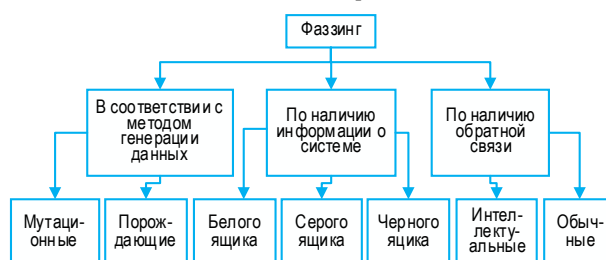


Рис. 1 Классификация видов фаззинга

Fig. 1. Classification of Fuzzing Methods

При тестировании протоколов в случае неизвестной структуры и правил взаимодействия – применяется методология «черного ящика» (BB – аббр. от англ. Black Box), при наличии информации о логической и процедурных характеристиках протокола – методы «серого» (GB – аббр. от англ. Grey Box) и «белого» (WB – аббр. от англ. White Box) ящика. Наличие обратной связи является ключевым признаком проверки функционирования протокола или выяснения степени влияния на протокол тестовой конструкции.

В работах [3, 4] выделены основные функциональные особенности, характерные для программных фаззеров, а также характеристики, связанные с обработкой протоколов разных типов, возможности формирования структуры пакета и генерации тестовых последовательностей, наличие исходного кода. Развитие выделенных программных средств идет с расширением их функционала, в том числе и в области работы с протокольными конструкциями, а также в направлении качества проведенных тестовых исследований, за счет применения современных математических методов: марковских цепей, машинного обучения, генетических алгоритмов.

В таблице 1 проведен анализ функциональных характеристик фаззеров и их классификация, в соответствии с введенными ранее на рисунке 1 признаками. Функциональные возможности фаззеров с возможностью описания структуры пакета в соответствии с вышеприведенной классификацией считаются характерным признаком порождающих типов.

ТАБЛИЦА 1. Анализ фаззеров в приложении к анализу протоколов

TABLE 1. Fuzzer Analysis in an Application to Protocol Analysis

Фаззер, тип [ссылка]	Генерация пакетов	Язык	Поддержка протоколов	Описание структуры пакета	Мутационные без описания структуры	Тип обратной связи
AFL, GB [http://lcamtuf.coredump.cx/afl]	–	C, C++	–	–	Нет	Анализ хода выполнения программы
Peach Fuzzer, WB [https://www.peach.tech]	+	XML, C#	Не ограничен	+	Да	Анализ хода выполнения программы
Honggfuzz, GB [https://github.com/google/honggfuzz]	–	C	–	–	Нет	Анализ хода выполнения программы
Boofuzz, GB [https://boofuzz.readthedocs.io/en/stable]	+	Python	UDP, TCP	+	Нет	Анализ ответа
Sulley, GB [http://www.fuzzing.org/wp-content/SulleyManual.pdf]	+	Python	UDP, TCP	+	Нет	Анализ ответа
Radamsa, BB [https://gitlab.com/akihe/radamsa]	+	C	Не ограничен	–	Да	Нет
Cert BFF, GB [https://github.com/CERTCC/certifuzz]	–	Python	–	+	Да	Анализ хода выполнения программы
Pulsar, GB [7]	+	Python	Не ограничен	+	Да	Анализ ответа
AFLNet, GB [https://github.com/aflnet/aflnet]	–	C	Не ограни- чен	–	Да	Анализ хода выполнения программы
Netzob, GB [https://github.com/netzob/netzob]	+	Python	Не ограничен	+	Да	Анализ ответа
ProtoFuzz, GB [https://github.com/trailofbits/prototofuzz]	–	Python	–	+	Нет	Анализ тестируемого объекта
Doona, GB [https://github.com/wireghoul/doona]	+	Perl	Не ограничен	+	Нет	Наличие ответа

Мутационные фаззеры на основе известных структурных конструкций пакетов порождают новые тестовые последовательности, применяемые далее в работе.

По наличию обратной связи программные средства разделяются по способу контроля хода выполнения программы следующими функциональными возможностями:

- "Анализ хода выполнения программы" и "Анализ ответа";
- "Анализ тестируемого объекта" и "Наличие ответа".

Из таблицы видно, что часть существующих решений разработана не только для анализа ПО, но и для анализа протоколов.

Программы без возможности генерации пакета могут осуществлять частичное тестирование уязвимостей телекоммуникационных протоколов. Например, во время анализа исходного кода могут быть затронуты участки, отвечающие за обработку трафика, что порождает отправку или получение пакета.

Большая часть телекоммуникационных фаззеров сосредоточена на тестировании высокоуровневых текстовых протоколов и назвать их универсальными нельзя. Таким образом для тестирования протоколов и групп протоколов лучшими функциональными характеристиками обладают: Radamsa, Pulsar, Netzob. Выделенные средства для тестирования вне зависимости от применяемого протокола обладают возможностью формирования сетевых пакетов и мутационным подходом к изменению тестовых данных. Фаззеры с обратной связью (в виде анализа хода выполнения программы) имеют ограничения, связанные с универсальностью применения. В случае наличия обратной связи для каждой разновидности платформы требуется разработка агента для обратной связи, что усложняет реализацию.

Фаззеры Pulsar и Netzob формируют данные на основе изменения графа модели взаимодействия протокола. Эти программы анализируют входной трафик, формируют изменения графа модели взаимодействия протокола с помощью математиче-

ских методов кластеризации, машинного обучения и марковских моделей. Основное внимание в данных фаззерах уделено моделированию процедурной характеристики работы протокола, а не изменению структуры пакетов.

Инструмент для генерации и мутации входных данных – Radamsa – предназначен исключительно для работы с входными данными и не предназначен для получения ответов от сервера. Однако он сам по себе не имеет функциональности для отправки пакетов или взаимодействия с серверами.

Таким образом, из результатов анализа следует, что универсального средства тестирования протоколов, не требующего знания о структуре протокола, не существует, что говорит об актуальности задачи формирования фаззера для протокола или группы протоколов. Направление развития программных фаззеров является применение математических методов для снижения рутинных функций инженера, а также разработка способов автоматизации процесса тестирования, что также может учтено при развитии решений, предназначенных для проверки протокольных конструкций.

Фаззинг протоколов

Проанализированные функциональные возможности средств фаззинга протоколов, представленные в таблице 1, позволяют дополнить уже существующую классификацию (рисунок 1) новыми классификационными признаками, представленными на рисунке 2.



Рис. 2. Классификация видов фаззинга в соответствии с целевой областью телекоммуникаций

Fig. 2. Classification of Fuzzing Types According to the Target Area of Telecommunications

В зависимости от уровня модели взаимодействия открытых систем (ЭМВОС), на котором находится объект тестирования, принято выделять фаззеры протоколов и веб-приложений.

Под фаззерами протоколов принято понимать программы, взаимодействующие на соответствующих уровнях стека коммуникационных протоколов. В настоящий момент существуют следующие подходы к их реализации: в фаззере протоколов Peach Fuzzer, в специальном файле конфигурации заранее описывается структура пакета и обозначаются типы данных для каждого из них. В ходе тестирования создаются пакеты согласно шаблону. В данном решении в качестве метрики тестирования используются коды завершения программы, получаемые с помощью дебаггера на сервере.

Схема работы фаззера показана на рисунке 3, особое внимание следует обратить на то, что данный инструмент используется преимущественно для работы протоколами прикладных уровней, а транспортировку тестовых конструкций осуществляют с помощью протоколов TCP, UDP. Это ограничивает сферу его применения протоколами прикладного уровня.

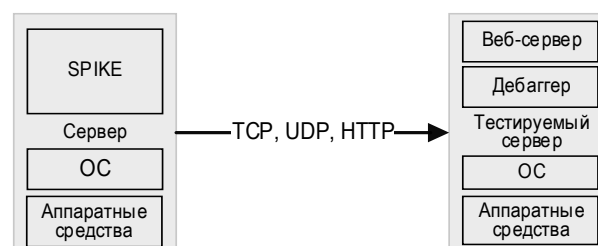


Рис. 3. Схема работы генерационного фаззера

Fig. 3. Schematic Diagram of the Generating Fuzzer Operation

С точки зрения протокольных программ для тестирования отдельные исследования [7] выделяют методы генерации пакетов, применяемые в AFLNet, которые могут быть реализованы в программах тестирования протоколов. В фаззере протоколов AFLNet применяется метод генерации программного фаззера AFL, где с помощью средств операционной системы имитируется получение тестируемым приложением пакетов. Таким образом, фаззер протоколов AFLNet имитирует функции протокольного фаззера на основе функции программного фаззера.

При генерации данных мутационным фаззером выделяются следующие простейшие операции, порождающие изменение входных данных – мутацию: вставка (случайного) символа; удаление символа; инверсия бита в представлении символа; случайный выбор мутации.

Помимо операций над битами, рассмотренные действия могут быть применены к другим единицам данных, например, в программном средстве AFL, помимо вышеописанных действий над битами, аналогичные операции применяются и к байтам.

Как и в AFL, в качестве метрики для фаззинга используется покрытие кода. Чем большая часть кода затронута в ходе тестирования, тем более безопасной считается реализации протокола в программе. Схема реализации процесса тестирования представлена на рисунке 4.

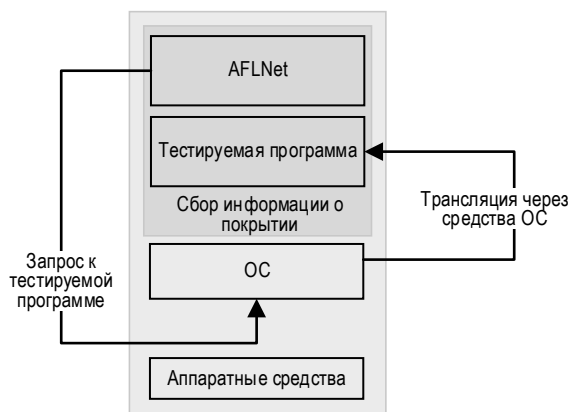


Рис. 4 Схема работы мутационного фаззера с трансляцией средствами ПО

Fig. 4. Scheme of Operation of Mutation Fuzzers with Translation by Software Means

Особенностью фаззера AFLNet является то, что создаваемые для тестирования пакеты с помощью средств ОС транслируются на вход тестируемой программы, установленной на той же ЭВМ. При реализации данных методов в качестве объекта тестирования всегда рассматривается программа, функционирующая на 7 уровне ЭМВОС.

Метод тестирования протоколов

В случае телекоммуникационного оборудования, тестирование может быть реализовано как при пересылке трафика, так и при обработке служебных (сигнальных) протоколов. При обработке полученного трафика ошибка может вызвана не только в программных модулях различных уровнях модели ЭМВОС, но и в системном ПО, например – в сетевых драйверах.

Для тестирования всех потенциально уязвимых протоколов сетевого взаимодействия, предлагается новый метод тестирования, предполагающий не только выявление ошибок в системном ПО, но и в конфигурации телекоммуникационного оборудования и иных компонентах обработки трафика, вместе формирующих сетевую коммуникационную инфраструктуру объекта КИИ.

При реализации данного подхода предлагается располагать устройство с фаззером в точке максимальной концентрации трафика в сетевом фрагменте или создавать такую точку, например, на основе технологии зеркалирования порта (Switch port analysis) и создавать множество (корпус) тестов из уже сгенерированного трафика, тем самым решая задачу формирования исходных параметров для тестирования.

Предполагается, что если пакеты, взятые за основу, сформированы согласно правилам (валидны), то и порожденные в ходе фаззинга пакеты с большей вероятностью не будут отброшены. Помимо этого, используя трафик системы в качестве исходного, нет необходимости производить его мониторинг для получения списка узлов и тестируемых служб. В этом случае достаточно лишь не изменять поля отправителя и получателя пакета, тогда сформированный трафик дойдет до получателя.

Для автоматизации процесса генерации тестовых фреймов и пакетов предполагается работать с заголовками пакетов в качестве базовых единиц. В связи с данными предложениями, предлагается модифицировать их структуру и организовать процесс генерации согласно рисунку 5. Получив исходное множество для генерации параметров (популяции), фаззер, согласно генетическому алгоритму, выбирает некоторое подмножество (популяции) и применяет к ним операцию объединения (кроссовера), объединяя их информацию за исключением заголовков, необходимых для продвижения пакета по сети (IP-адрес источника и назначения в примере на рисунке 5).

Определяется поле протокола, для которого будет выполнена операция мутации. На рисунке 5 мутация применена к заголовку смещения. В качестве пакетов для скрещивания выбирается IP-пакет С и IP-пакет D из множества IP-пакетов А, В, С, D. В качестве места объединения выбрано число 6, первые 6 параметров заголовка пакета С объединяются с оставшимися заголовками пакета D. Случайным образом принимается решение о необходимости мутации и, в качестве мутируемого заголовка, выбирается поле "Смещение" (рисунок 5).

Сгенерированный пакет отправляется тестируемой системе, причем для реализации данной операции необходимо выбирать программные средства, позволяющие создавать «неправильные» по спецификации пакеты (изменение порядка следования заголовков, несоответствие фактической длины пакеты и значения поля).

Эмпирической функцией эффективности предполагается выбрать время ответа, следовательно, чем сильнее «нагрузил» пакет систему, тем более он эффективен.

Уточним, что продемонстрированный метод применим не только к IP-пакетам (сетевому уровню), он и к единицам данных любого уровня, что демонстрируется на рисунке 6, где представлен процесс тестирования на основе единиц, передаваемых данных различного типа – пакетов и фрейма. В ходе операции кроссовера MAC-адрес назначения IP-пакета заменяется на адрес фрейма, а операции мутации подвергается поле инкапсулированного протокола.

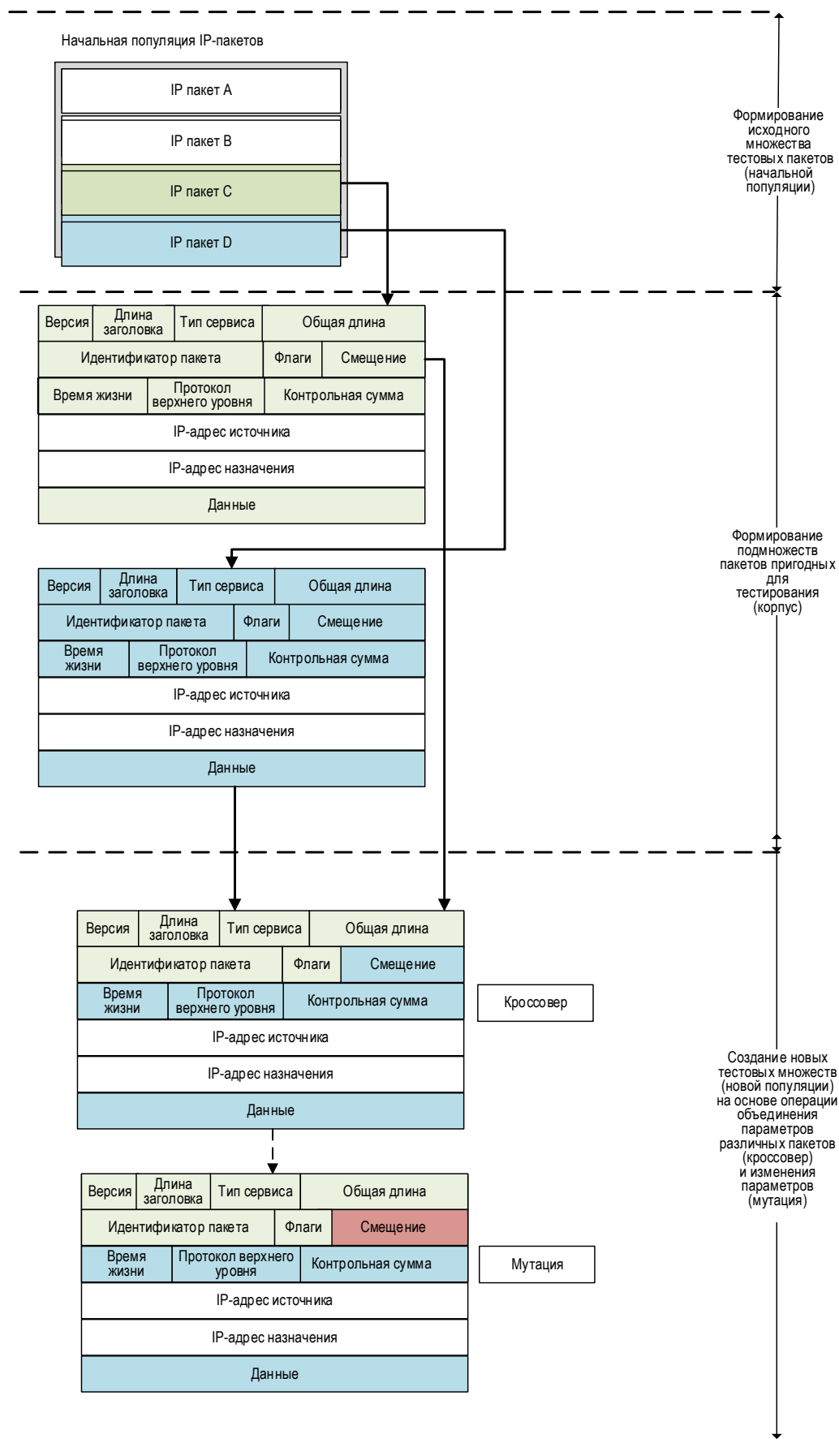


Рис. 5. Процесс генерации данных согласно эволюционному алгоритму на сетевом уровне

Fig. 5. The Process of Data Generation According to Evolutionary Algorithm at Network Layer

Предлагаемый на рисунке 6 метод тестирования протоколов или групп протоколов на основе генетического алгоритма, на примере протоколов VTP, IP.

Пусть из множества пакетов перехваченного в точке концентрации трафика случайным образом выбирается IP-пакет – D и VTP фрейм (рисунок 6). На следующем шаге выбирается случайное число размером меньшей единицы данных – VTP фрейма, на рисунке 6 выбрано число 1.

Следующим шагом производится процедура кроссовера, для этого от VTP фрейма выбирается только 1 заголовок (согласно ранее выбранному числу, (выделено зеленым цветом на рисунке 6), остальные заголовки формируемой тестовой единицы данных копируются у IP-пакета – D).

При формировании нового пакета не могут быть взяты в качестве точки конкатенации адресные заголовки сетевого уровня для обеспечения доставки пакета существующей службе.

На следующем этапе случайным образом выбирается, будет ли применена мутация. На приведенном примере мутация применяется к случайно выбранному заголовку (инкапсулированный протокол), изменяя его значение. В качестве мутации могут быть применены как алгоритмы мутации, реализованные в программном средстве AFL, так и иные алгоритмы.

Алгоритм реализации универсального программного средства тестирования протоколов или групп протоколов, на основе генетических методов представлен на рисунке 7.

Описание алгоритма генерации данных согласно эволюционному алгоритму

Перехват трафика, формирования корпуса тестирования. В качестве точки перехвата видится необходимым выбирать точку сети с максимальной концентрацией трафика (коммутатор, маршрутизатор, межсетевой экран), либо формировать такую точку самостоятельно.

В ходе дальнейшего тестирования в исходном множестве тестовых пакетов будет присутствовать максимальное разнообразие реального трафика системы. При генерации данных наличие большого количества трафика позволит добиться увеличения числа тестируемых сервисов и устройств (при применении процедуры кроссовера или мутации, при этом поля заголовков источника и назначения не затрагиваются).

Выбор 2-х единиц данных – исходного множества для формирования тестовых пакетов, случайным образом или в соответствии с некоторой метрикой (например, временем ответа). В случае использования метрики для реализации генетического фаззера следует рассматривать функцию случайного выбора с заданными весами.

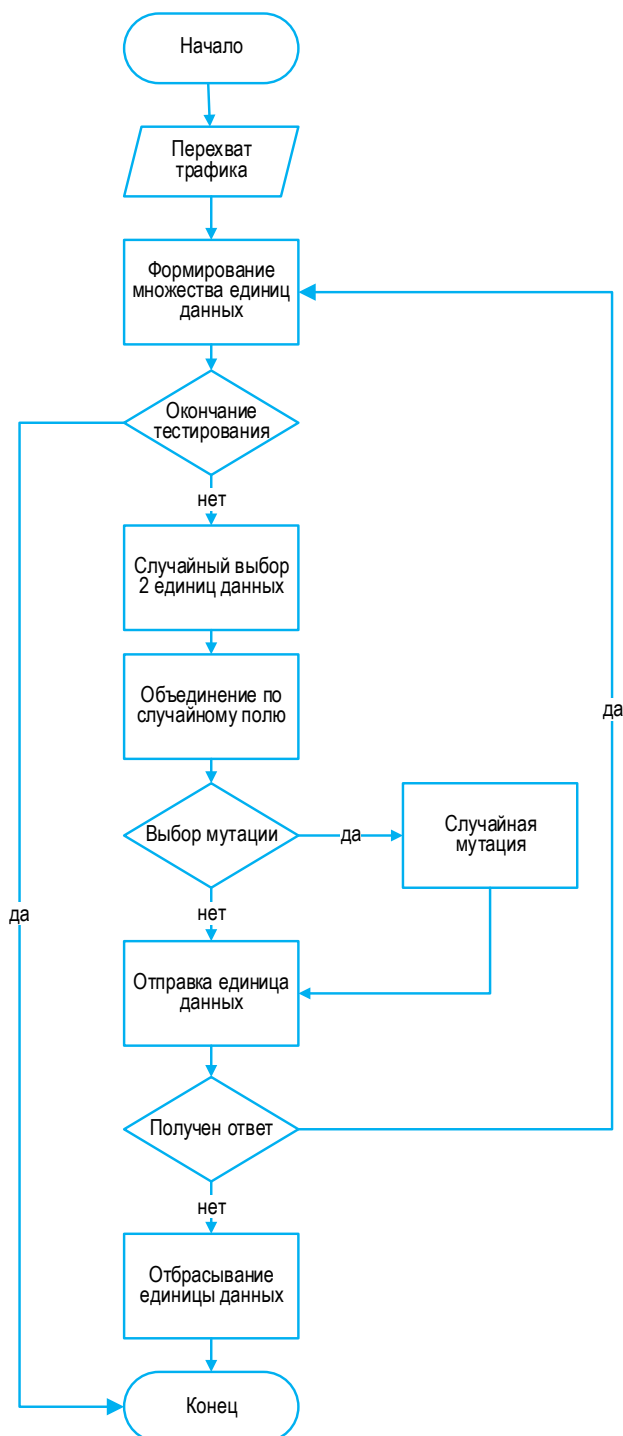


Рис. 7. Алгоритм генерации данных согласно эволюционному алгоритму

Fig. 7. Data Generation Algorithm According to Evolutionary Algorithm

Выбор номера случайного поля и объединение единиц данных соответственно. Реализация данной операции возможна как с учетом уровня протокола согласно ЭМВОС, так и без нее. Необходимо принять решение о объединении только общих заголовков или объединении без учета структуры каждой из единиц исходных данных.

Случайный выбор мутации. Мутация и решение о ее применении выбираются случайным образом. В качестве мутаций, в отличие от AFL, предлагается выбирать не универсальные операции над байтами, а основанную на типе данных генерацию потенциально некорректных значений. Примером реализации такого подхода является фреймворк Scapy (<https://scapy.readthedocs.io/en/latest/usage.html#fuzzin>).

Отправка сгенерированного пакета. В качестве потенциальной метрики эффективности фаззера, предлагаемого к реализации, рекомендуется выбрать время ответа сервера, которое, при задании известной задержке передачи, характеризует время обработки пакета сервером. Если же пакет вызовет отказ в обслуживании, данное явление будет возможно вычислить по отсутствию дальнейших ответов на запрос. При реализации тестирования методом черного ящика предлагаем периодически проверять состояние тестируемого сервиса или устройства посредством протокола ICMP.

Помещение пакета в корпус или отбрасывание. При генерации валидного пакета следует поместить его в корпус тестирования для увеличения потенциального разнообразия данных для генетического алгоритма. При использовании метрики предлагаем в качестве ее рассматривать значение времени ответа, нормированное относительно задержки передачи в сети.

Одним из направлений развития алгоритма является применение математического аппарата Марковских цепей для осуществления случайной (стохастической) выборки, осуществляемой по нормальному (равномерному) закону распределения. Такими элементами случайной выборки могут быть протоколы, группы протоколов, параметров протоколов для изменения (мутации), которые являются последовательностью дискретных случайных величин, выражение (1), где X_n – множество дискретных величин, протоколов, групп протоколов, параметров протоколов телекоммуникационных систем.

$$P(X_n = x_n | X_{n-1} = i_{n-1}, X_{n-2} = x_{n-2}, \dots, X_0 = x_0) = P(X_n = x_n | X_{n-1} = x_{n-1}). \quad (1)$$

Матрица переходных вероятностей формируется на первоначальном этапе исследований на основе нормального закона, равномерного закона распределения, в дальнейшем экспертным способом, при необходимости исследования известных векторов атак в соответствии с известными угрозами, например, в соответствии с банком данных угроз безопасности информации ФСТЭК. Предпочтительный вариант решения формируется на основе экспериментальных исследований.

Заключение

В статье рассмотрены вопросы поиска уязвимостей в телекоммуникационных протоколах объектов КИИ. В ходе анализа установлено, что ближайшими аналогами таких средств является средства тестирования программ на уязвимости – программный фаззинг. Проведена классификация существующих средств и видов фаззинга, на ее основе установлены функциональные возможности существующих решений. Перспективное направление развития таких средств связано с применением математических методов для автоматизации процесса тестирования и поиска уязвимостей.

В ходе исследования предметной области телекоммуникаций были выявлены недостатки реализации существующих решений, связанные с экспертным подходом, недостаточным уровнем применения математических методов при формировании тестовых комбинаций. Представлены предложения по построению алгоритма функционирования универсального фаззера протоколов на основе математического аппарата, генетических алгоритмов, Марковских цепей.

Результатом исследования стал разработанный способ генерации тестовых данных для проверки телекоммуникационных объектов, не требующий предварительной настройки и не зависящий от протокольной иерархии объекта тестирования.

Предполагается, что предложенный метод в ходе тестирования затронет все уровни ЭМВОС, несколько устройств для обработки трафика. Направлением развития предлагаемого решения является разработка средств автоматического, автоматизированного поиска уязвимостей в протокольных конструкциях объектов КИИ.

Список источников

1. Девянин П.Н., Тележников В.Ю., Хорошилов А.В. Формирование методологии разработки безопасного системного программного обеспечения на примере операционных систем // Труды Института Системного Программирования РАН. 2021. Т. 33. № 5. С. 25–40.
2. Вареница В.В., Марков А.С., Савченко В.В. Практические аспекты выявления уязвимостей при проведении сертификационных испытаний программных средств защиты информации // Вопросы Кибербезопасности. 2021. № 5 (45). С. 36–44.

3. Манеев А.О., Спивак А.И. Стохастическое тестирование программного обеспечения для поиска уязвимостей // Научно-технический вестник информационных технологий, механики и оптики. 2021. Т. 21. № 6. С. 895–902.
4. Козачок А.В., Козачок В.И., Осипова Н.С., Пономарев Д.В. Обзор исследований по применению методов машинного обучения для повышения эффективности фаззинг-тестирования // Вестник Воронежского государственного университета. Серия: системный анализ и информационные технологии. 2021. №4. С. 83–106
5. Саттон М., Амини П., Грин А. Fuzzing: исследование уязвимостей методом грубой силы. Пер. с англ. М.: Символ-Плюс, 2017. 554 с.
6. Gascon H., Wressnegger C., Yamaguchi F., Arp D., Rieck K. PULSAR: Stateful Black-Box Fuzzing of Proprietary Network Protocols // Proceedings of the 11th International Conference on Security and Privacy in Communication Networks (SecureComm, Dallas, USA, 26–29 October 2015). Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Cham: Springer, 2015. Vol. 164. PP. 330–347. DOI:10.1007/978-3-319-28865-9_18
7. Шарков И.В., Падарян В.А., Хенкин П.В. Об особенностях фаззинг-тестирования сетевых интерфейсов в условиях отсутствия исходных текстов // Труды Института Системного Программирования РАН. 2021. Т. 33. № 4. С. 211–226.

References

1. Devyanin P.N., Telezhnikov V.Yu., Khoroshilov A.V. Formation of methodology of safe system software development on the example of operating systems. *Proceedings of ISP RAS*. 2021;33(5):25–40.
2. Varenitsa V.V., Markov A.S., Savchenko V.V. Practical aspects of vulnerability detection during certification testing of information protection software. *Voprosy Kiberbezopasnosti*. 2021;5(45):36–44.
3. Maneev A.O., Spivak A.I. Stochastic Software Testing for Vulnerability Analysis. Scientific and Technical Journal of Information Technologies, Mechanics and Optics, Mechanics and Optics. 2021;21(6):895–902.
4. Kozachok A.V., Kozachok V.I., Osipova N.S., Ponomarev D.V. Overview of Studies on the Application of Machine Learning Methods to Improve the Efficiency of Fusing Testing. *Proceedings of Voronezh State University. Series: Systems Analysis and Information Technologies*. 2021;4:83–106.
5. Sutton M., Amini P., Green A. *Fuzzing: vulnerability research by brute force method*. Translated from English. Moscow: Symbol-Plus Publ.; 2017. 554 p.
6. Gascon H., Wressnegger C., Yamaguchi F., Arp D., Rieck K. PULSAR: Stateful Black-Box Fuzzing of Proprietary Network Protocols. *Proceedings of the 11th International Conference on Security and Privacy in Communication Networks, SecureComm, 26–29 October 2015, Dallas, USA. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol.164*. Cham: Springer; 2015. p.330–347. DOI:10.1007/978-3-319-28865-9_18
7. Sharkov I.V., Padarian V.A., Henkin P.V. Features of Fuzzing Network Interfaces Without Source Codes. *Proceedings of ISP RAS*. 2021;33(4):211–226.

Статья поступила в редакцию 21.11.2023; одобрена после рецензирования 29.11.2023; принята к публикации 06.12.2023.

The article was submitted 21.11.2023; approved after reviewing 29.11.2023; accepted for publication 06.12.2023.

Информация об авторах:

ВАСИНЕВ Дмитрий Александрович	кандидат технических наук, доцент, сотрудник Академии ФСО России  https://orcid.org/0009-0004-7030-5421
СОЛОВЬЕВ Михаил Викторович	сотрудник Академии ФСО России  https://orcid.org/0009-0006-6472-1167

Научная статья

УДК 004.42

DOI:10.31854/1813-324X-2023-9-6-68-82



Методология проведения реверс-инжиниринга машинного кода. Часть 2. Статическое исследование

 Константин Евгеньевич Израилов, konstantin.izrailov@mail.ru

Санкт-Петербургский Федеральный исследовательский центр Российской академии наук,
Санкт-Петербург, 199178, Российская Федерация

Аннотация: Изложены результаты создания единой методологии проведения реверс-инжиниринга машинного кода устройств. Данная, вторая часть цикла статей посвящена статическому исследованию кода с целью восстановления его метаданных (исходного кода, алгоритмов, архитектуры, концептуальной модели), а также поиска в нем уязвимостей. Проводится обзор научных публикаций на тему существующих методов и средств статического анализа машинного кода. Дается детальное описание и формализация шагов этапа, а также примеры их применения на практике. Частичная схема предлагаемой методологии приводится в графическом виде с указанием получаемых основных и промежуточных результатов.

Ключевые слова: реверс-инжиниринг, обратная разработка, программная инженерия, статический анализ, информационная безопасность, уязвимости, методология, схема

Ссылка для цитирования: Израилов К.Е. Методология реверс-инжиниринга машинного кода. Часть 2. Статическое исследование // Труды учебных заведений связи. 2023. Т. 9. № 6. С. 68–82. DOI:10.31854/1813-324X-2023-9-6-68-82

Methodology for Machine Code Reverse Engineering. Part 2. Static Investigation

 Konstantin Izrailov, konstantin.izrailov@mail.ru

Saint-Petersburg Federal Research Center of the Russian Academy of Sciences,
St. Petersburg, 199178, Russian Federation

Abstract: The creating results a unified methodology for reverse engineering the machine code of devices are presented. This second part of the articles series is devoted to static research of code in order to restore its metainformation (source code, algorithms, architecture, conceptual model), as well as search for vulnerabilities in it. A scientific publications review on the topic of existing methods and tools for static analysis of machine code is carried out. A detailed description and formalization of the steps of the stage is given, as well as examples of their application in practice. A proposed methodology partial diagram is presented in graphical form, indicating the main and intermediate results obtained.

Keywords: reverse engineering, backwards engineering, software engineering, static analysis, information security, vulnerabilities, methodology, scheme

For citation: Izrailov K. Methodology for Machine Code Reverse Engineering. Part 2. Static Investigation. *Proceedings of Telecommun. Univ.* 2023;9(6):68–82. DOI:10.31854/1813-324X-2023-9-6-68-82

Введение

Статья данного цикла посвящена актуальной задаче – реверсу-инжинирингу (далее – РИ) машинного кода (далее – МК), обусловленной необходимостью восстановления принципов и деталей работы программного обеспечения при отсутствии его исходного кода (далее – ИК). Результаты такого исследования оказываются востребованы в следующих случаях. Во-первых, отличие функционала МК от заявленного может означать наличие в нем уязвимостей, потенциально приводящих к информационным угрозам. Во-вторых, потеря ИК программной библиотеки в ряде случаев может затруднить разработку взаимодействующих с ней продуктов, для чего требуется его хотя бы частичное восстановление. В-третьих, одним из способов импортозамещения (естественно, с учетом юридических сторон вопроса) может стать глубокое и всестороннее изучение МК зарубежных продуктов и разработка на их основе отечественных аналогов.

В первой части цикла статей [1] было дано общее представление о предлагаемой автором методологии РИ (далее – Методология), для описания которой вводилась онтологическая модель предметной области. Также был описан 1-й из 4 этапов Методологии, посвященный подготовке объекта исследования, под которым понимается МК всех программ из единой программной системы. Продолжая представление Методологии, во второй части цикла будет описан ее 2-й этап, посвященный наиболее трудоемкой и технически сложной части исследования – статическому анализу (т. е. без выполнения программ), основной задачей которого считается восстановление различной человеко-ориентированной метаинформации, потерянной при компиляции исходного кода в МК.

Обзор работ

Проведем обзор работ, посвященных вопросу РИ в части существующих методов и средств для статического анализа МК; при этом уделим внимание именно отдельным шагам такого анализа, их принципам и особенностям. Все это позволит построить этапы Методологии, относящиеся к статическому исследованию МК.

Работа [2] посвящена обзору различных методов и программных средств, предназначенных для анализа МК в интересах восстановления алгоритмов его работы, а также используемых форматов данных. Отмечается некоторая успешность решения задачи РИ с помощью продукта IDA Pro и различных отладчиков МК, хотя и с некоторыми ограничениями в виде работы с небольшими участками бинарных данных, а также сложности автоматизации процесса. Авторами предложено применение методов статического анализа совместно с динамическими, что дает более полную картину функционала исследуемой программы.

Отдельное внимание в статье уделяется восстановлению формата данных, под которым понимаются границы полей в программном объекте, их группировка, семантика и дополнительная информация (константные значения, связи между полями-указателями и пр.). Часть авторского коллектива в другой работе [3] описывает применение статико-динамического анализа МК для поиска в нем недекларированных возможностей.

В [4] делается сравнительный анализ методов и средств декомпиляции программ (т. е. восстановления их ИК по машинному коду), а также предлагается новый метод определения типов данных в МК. Рассматриваются такие декомпиляторы, как Boomerang, DCC, REC и Hex-Rays (как плагин к IDA Pro). Согласно балльному сравнению, наилучшими показателями обладает Hex-Rays, хотя все утилиты достаточно плохо справляются с задачей корректного восстановления типов данных. В качестве альтернативы авторы описывают разрабатываемую среду декомпиляции, гипотетически (и отчасти на практике) имеющую преимущества среди указанных конкурентов.

В статье [5] описывается метод восстановительной коррекции МК после его дизассемблирования, который с помощью эмуляторов исправляет некоторые ошибки процесса. В процессе анализа МК представляется с помощью графов управления, схем Янова и сетей Петри. Затем, с помощью ряда метрик определяются участки кода, подверженные обфускации, а также имеющие чрезмерно сложную структуру. После этого выявляются потенциально опасные нарушения логики вычислений, заикливания и «мертвый код». Также осуществляется поиск недекларированных возможностей на основании ряда признаков.

В работе [6] исследуется возможность разбиения МК на отдельные логически связанные блоки в автоматическом режиме, что, в том числе, позволит выбрать вектор исследования кода (как последовательности переходов между его инструкциями) в интересах поиска уязвимостей. Авторы указывают на необходимость определения в процессе статического анализа точек входа в программу и входных данных. Также приводятся варианты использования графов вызовов подпрограмм и их матрицы смежности. С каждым типом уязвимости предлагается связать группу матриц, содержащих количество потенциально опасных участков МК.

Исследование [7] направлено на обнаружение вредоносного кода путем РИ МК. Приводится обобщенная схема РИ, состоящая из статического и динамического анализа, первый из которых осуществляется с применением дизассемблеров (например, IDA Pro), а второй – отладчиков (например, OllyDebugger).

Описывается утилита PEiD, анализирующая программу на предмет определения языка программирования, компилятора, упаковщика и другой информации о ней. Если программа оказывается запакованной, то предлагается ее распаковать путем динамического анализа, а затем применить классический статический анализ для поиска уязвимостей.

Согласно сделанным обзорам, существует достаточно ограниченное количество публикаций, посвященных непосредственно статическими методами и средствам обработки МК, и восстановлению из него информации, подходящей для экспертного анализа; при этом, в основном, из МК рассматривается возможность получения ассемблерного листинга, форматов данных, а также частично псевдо-ИК и различных графических схем его алгоритмов. Авторские же публикации в данной области среди всех занимают существенное место и направлены на восстановление большего объема метаинформации о МК, включающей также достаточно высокоуровневую – архитектуру и концептуальную модель программы (или их системы).

Схема Методологии

Онтологическая модель

Для точности описания Методологии ранее в [1] была предложена онтологическая модель предметной области; опуская ее детали и графическую схему, кратко опишем введенные таким образом основные сущности (которые далее будут писаться с большой буквы):

- 1) Машинный код – представление логики работы программ в виде инструкций процессора;
- 2) Программа – бинарный файл с МК и дополнительной информацией;
- 3) Программная система – совокупность взаимодействующих Программ;
- 4) Образ – представление Программной системы в виде монолитного файла;
- 5) Устройство хранения – устройство для получения и выполнения загруженного Образа;
- 6) Метаинформация – сведения о функционале МК (псевдокод, алгоритмы, архитектура, концептуальная модель и т. п.);
- 7) Реверс-инжиниринг – процесс восстановления Метаинформации о МК;
- 8) Уязвимость – отличие реализации МК от задуманной или заявленной, приводящее к реализации угроз.

Таким образом, онтологическая модель охватывает всю область РИ, вплоть до уязвимостей в МК.

Графическое представление

Предлагаемая Методология состоит из 4 этапов, шаги 1-го из которых были детально расписаны в предыдущей статье цикла, а 2-й, наиболее объемный, будет описан в текущей. Как и раньше, иден-

тификаторы шагов этапов имеют следующую запись: $X.Y$, где X – номер этапа, Y – порядковый номер выполнения в рамках этапа. Схема Методологии с отражением 2-го этапа в графическом виде представлена на рисунке 1.

На схеме используются следующие обозначения: прямоугольник с белым фоном – результат применения шага; круг со светло-желтым фоном – идентификатор шага в текущей статье цикла; круг с темно-желтым фоном – идентификатор шага в предыдущей статье цикла; сплошная стрелка – основное действие шага над результатом предыдущего; пунктирная стрелка – дополнительное использование шагом результатов другого шага; прямоугольники с закругленными краями и надписью – группы логически связанных элементов (синий фон – для текущего этапа, серый фон – для последующего этапа); пунктирный прямоугольник с надписью – область шагов определенного этапа. Так, область «Этап 1. Подготовка объекта исследования» соответствует шагам (кроме 2.7 и 2.8), описанным в предыдущей статье цикла, область «Этап 2, 3. Анализ объекта исследования» – шагам, описанным в текущей (для статического исследования) и последующей (для динамического исследования) статьях, область «Этап 4. Документирование» – шаги также для последующей статьи цикла.

Согласно схеме (см. рисунок 1), в данной части статьи будут расписаны Шаги с 2.1 по 2.14; они полностью определяют 2-й этап РИ и предназначены для анализа МК статическими способами. Опишем далее эти шаги более детально.

Этап 2. Статическое исследование

Этап предназначен для выполнения основного анализа МК без его непосредственного выполнения – что отражает статичность исследования. Для каждого шага будет дан пример его применения на практике, используя, по возможности, рассмотренный в предыдущей статье цикла DXE-драйвер UEFI-фазы *BDS* (аббр. от англ. Boot Device Selection, *перев. на русс.* Выбор Загрузочного Устройства) [8] с именем *BdsDxe.efi* из материнской платы *PRIME Z490-A* производства *ASUSTeK COMPUTER INC.* Также, примеры будут считаться типовыми без учета специфических ситуаций, например, когда секции во вредоносной программе имеют заведомо неверный тип для противодействия антивирусным средствам.

Шаг 2.1. Анализ МК для выделения точек входа, секций кода и данных

Шаг является первым для этапа статического исследования, поскольку с него начинается основная ветка РИ МК, полученного из каждой Программы, расположенной в Программной системе Образа, выполняемого на Устройстве хранения. Входом шага являются результаты применения Этапа 1.

Задачей шага является получение из программных секций с кодом и данными (далее – Секций) содержащегося в них МК. Поскольку Секции, выделенные на Шагах 1.7 и 1.8, имеют определенный тип, определяющий хранящиеся в них данные, то задачей шага является выбор нужных секций и экспорт из них МК.

Формальная запись шага ($Step_{2.1}$) имеет следующий вид:

$$\left\{ \begin{array}{l} MachineCodes = Step_{2.1}(Sections) \\ MachineCodes \equiv \{MachineCode_i\} \\ Sections \equiv \{Section_i\} \\ Class^{Section} \in \{Section^{Code}, Section^{Data}\} \end{array} \right.,$$

где $MachineCodes$ – множество блоков с МК, полученных из Секций на шаге; $MachineCode_i$ – i -й блок с МК из множества (для упрощения индекс i далее

опустим); $Sections$ – множество секций, полученных из исследуемой Программы на Шаге 1.8; $Section_i$ – i -я секция из множества (для упрощения индекс i далее опустим); $Class^{Section}$ – класс секции, которая содержит код ($Section^{Code}$) или данные ($Section^{Data}$).

Как правило, Секция с выполняемым кодом имеет название «.text», поэтому в большинстве случаев получение МК может заключаться в ее выборе в интерактивном дизассемблере.

Примером шага может быть открытие в продукте IDA Pro драйвера BdsDxe.efi, в результате чего отобразится 7 Секций (называемых также сегментами), одна из которых имеет имя «.text» и класс «CODE», и при переходе на которую будет отображаться содержимое ее МК (рисунок 2).

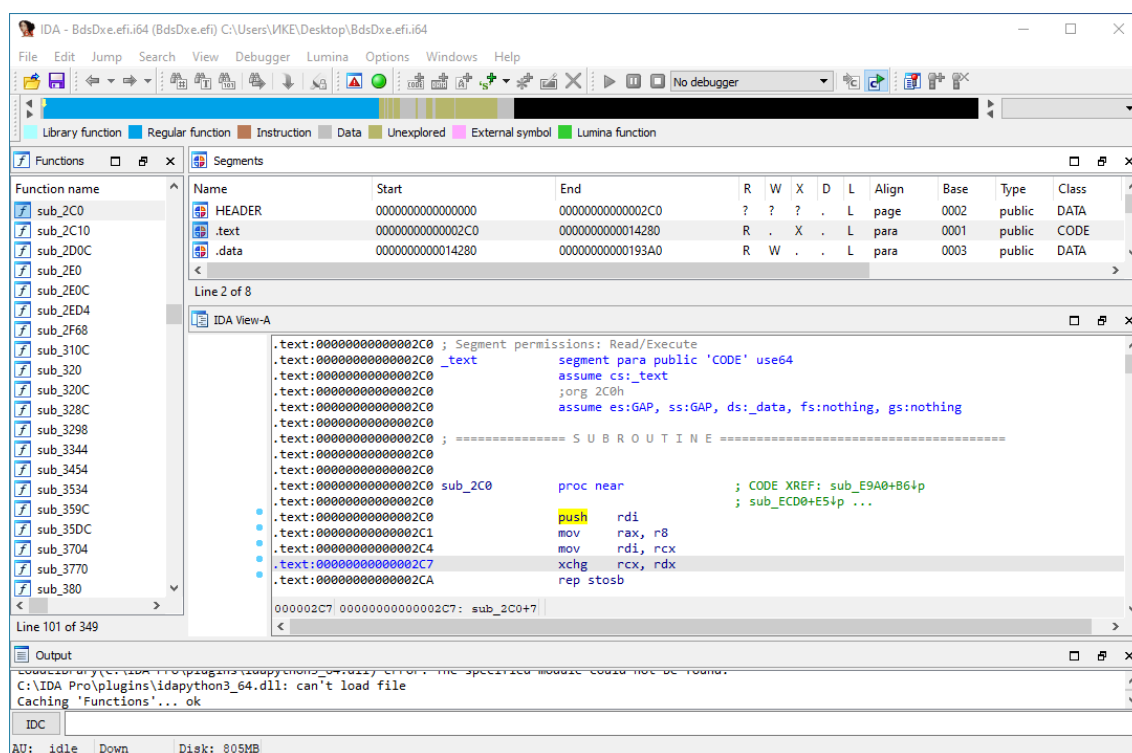


Рис. 2. Пример окон Segments и IDA View утилиты IDA Pro (для Секции кода)

Fig. 2. Segments and IDA View Windows Example of the IDA Pro Utility (for the Code Section)

Согласно рисунку 2, при выборе Секции «.text» с началом по 0x2C0 адресу, во втором окне отображается МК подпрограммы $sub_2C0()$, начинающейся с такого же адреса. Отметим, что точка входа в Программу – т. е. адрес первой выполняемой инструкции МК, также содержится в этой секции.

Шаг 2.2. Анализ секции данных для выделения глобальных данных

Задачей шага является получение из Секций, содержащихся в них глобальных данных, к которым могут быть отнесены такие объекты, как глобальные переменные, константы и пр. Аналогично

Шагу 2.1, каждая Секция имеет свой тип, что может быть учтено на данном шаге.

Формальная запись шага ($Step_{2.2}$) имеет следующий вид:

$$\left\{ \begin{array}{l} GlobalDatas = Step_{2.2}(Sections) \\ GlobalDatas \equiv \{GlobalData_i\} \end{array} \right.,$$

где $GlobalDatas$ – множество элементов глобальных данных, полученных из Секций на шаге; $GlobalData_i$ – i -й объект глобальных данных из множества (для упрощения индекс i далее опустим).

Как правило, основная Секция с глобальными данными имеет название «.data» (помимо дополнительных, таких, как «.rsrc» для ресурсов, «.xdata» для исключений и т. п.), поэтому в большинстве случаев их получение может заключаться в выборе нужной Секции в интерактивном дизассемблере.

Примером шага может быть продолжение исследования драйвера BdsDxe.efi в IDA Pro, для кото-

рого секция «.data» расположена по адресам 0x14280–0x193A и содержит, в том числе, следующие строковые константы (и их адреса):

EFI_WARN_UNKNOWN_GLYPH (0x14A10),
EFI_WARN_DELETE_FAILURE (0x14A2A),
EFI_WARN_WRITE_FAILURE (0x14A44);

отображение данной информации в IDA Pro представлено на рисунке 3.

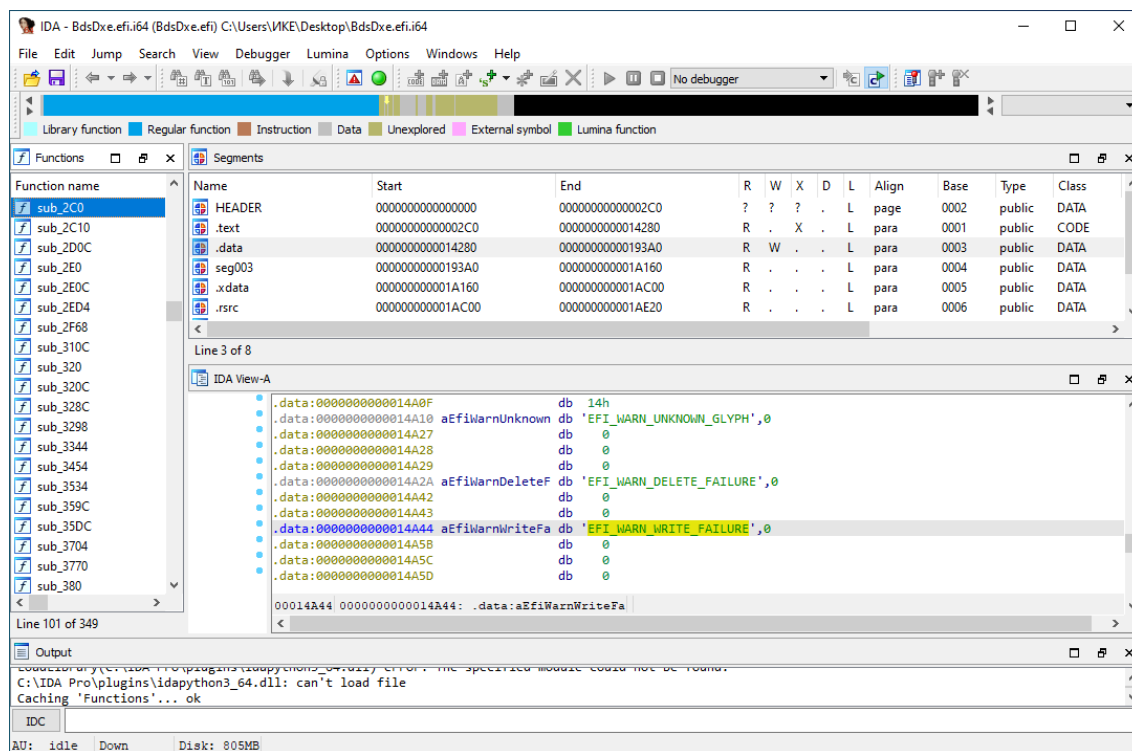


Рис. 3. Пример окон Segments и IDA View утилиты IDA Pro (для Секции данных)

Fig. 3. Segments and IDA View Windows Example of the IDA Pro utility (for the Data Section)

Согласно рисунку 3, при выборе Секции «.data» можно отобразить все глобальные данные Программы для их последующего анализа.

Шаг 2.3. Анализ МК для выделения подпрограмм

Шаг заключается в анализе МК для его деления на отдельные подпрограммы – т. е. многократно исполняемые участки инструкций; как правило, их вызов осуществляется путем перехода на первую инструкцию подпрограммы, а последняя инструкция содержит возврат в точку вызова. Традиционно, такие подпрограммы решают некоторую задачу, используя для

этого входные параметры и глобальные переменные, а также возвращая некоторый результат. Если подпрограмма возвращает данные, то она называется функцией, иначе – процедурой.

Формальная запись шага ($Step_{2.3}$), где $Subroutines$ – множество подпрограмм, полученных из МК на шаге; $Subroutine_i$ – i -я подпрограмма из множества (для упрощения, индекс i далее опустим); $Class_{Subroutine}$ – класс подпрограммы, соответствующий функции ($Subroutine^{Function}$) или процедуре ($Subroutine^{Procedure}$), имеет вид (1).

$$\begin{cases} Subroutines = Step_{2.3}(MachineCode) \\ Subroutines \equiv \{Subroutine_i\} \\ Class_{Subroutine} \in \{Subroutine^{Function}, Subroutine^{Procedure}\} \end{cases} \quad (1)$$

Задача выделения в МК подпрограмм не является тривиальной, хотя существуют достаточно эффективные ее решения, как правило, внедренные в

продукты для дизассемблирования. Примером шага может быть использование утилиты IDA Pro при анализе секций с МК. На рисунке 2 в левой

части экрана присутствует окно «Function name», содержащее все обнаруженные подпрограммы. Большинству из них даны автоматически сгенерированные имена в формате *sub_ADDR*, где *ADDR* – адрес начала подпрограммы (на рисунке 2 отображается их перечень по адресам с 0x2C0 по 0x380).

Шаг 2.4. Анализ МК выбранной подпрограммы для восстановления ее псевдокода

Шаг является наиболее сложным с научной и практической точки зрения, поскольку полноценного решения задачи восстановления ИК (или близких аналогов) из МК не существует. Основная причина этого заключается в том, что при компиляции – т. е. получении МК из ИК, основная часть Метаинформации, с которой работает программист (имена переменных и классов, логика выполнения, формулы вычислений, общая архитектура и т. п.), теряется. Восстановление же этой информации возможно по косвенным признакам; например, если в МК осуществляется многократный возврат выполнения на одну из предыдущих команд, то это может означать наличие логической конструкции языка программирования – *цикла*. Также, если множество подпрограмм расположены по близким адресам и делают взаимные вызовы друг другу, то это может означать наличие конструкции высокоуровневого языка – *экземпляра класса* (или даже логической конструкции архитектуры Программы – модуля).

Для улучшения качества восстановления псевдокода (т. е., по сути, его соответствия или компилируемости в МК) целесообразно использовать и глобальные данные, полученные на Шаге 2.2. Так, если в МК подпрограммы осуществляется отладочный вывод с помощью специальной процедуры *DebugPrint (name, message)*, где *name* – имя информирующего модуля, а *message* – сообщение с информацией, то передача в качестве параметров строковых констант из области глобальных данных позволит понять назначение этой подпрограммы. Например, если *name* ссылается на строку «LoginModule» (*перев. на русс.* Модуль авторизации), а *message* на «Invalid password» (*перев. на русс.* Неверный пароль), то с большой вероятностью подпрограмма, выводящая такую отладочную информацию, расположена в функционале по проверке пользовательского пароля при входе в систему. Формальная запись шага (Step 2.4), где *PseudoCode* – псевдокод, полученный из МК на шаге, имеет следующий вид:

$$PseudoCode = Step_{2.4}(Subroutine, GlobalData_s).$$

Соответственно, с помощью шага все части МК преобразуются в единый псевдокод, где *PseudoCodes* – множества псевдокода для всех подпрограмм МК; *PseudoCode_i* – *i*-й псевдокод из множества (для упрощения, *i* далее опустим):

$$PseudoCodes = \{PseudoCode_i\}.$$

Примером шага может быть применение достаточно редких и специализированных для этого утилит декомпиляции (т. е. осуществляющих обратную к компиляции операцию), таких, как плагин HexRays для IDA Pro [9], Ghidra [10] со встроенным декомпилятором или ряд разработок автора [11, 12]. Так, применение HexRays к подпрограмме *sub_1D88()* из DXE-драйвера *BdsDxe.efi*, имеющей ассемблерный код (в формате строк, где вначале идет адрес инструкции, а затем ее текстовая запись):

```
; Begin of sub_1D88()
0x1D88 sub    rsp, 38h
0x1D8C mov    edx, 1010000h
0x1D91 call   sub_6A8C
0x1D96 add    rsp, 38h
0x1D9A retn
; End of sub_1D88()
```

восстановит следующий псевдокод:

```
__int64 __fastcall sub_1D88(__int64 a1)
{
    return sub_6A8C(a1, 0x1010000i64);
},
```

соответствующий вызову другой подпрограммы *sub_6A8C()*, в которой помимо такого же 1-го параметра добавлен второй – 0x1010000.

Однако, поскольку у этих утилит есть ряд недостатков (включая работоспособность простых блоков МК, спорность понятности получаемого псевдокода, отсутствие поддержки сложных программных конструкций), шаг в основном выполняется вручную штучными высококвалифицированными экспертами.

Шаг 2.5. Анализ псевдокода выбранной подпрограммы для восстановления ее алгоритма

Шаг в той или иной степени имеет различные решения, поскольку получение алгоритма по ИК (а не только псевдокоду) считается достаточно частой задачей при анализе последнего. При этом, поскольку ИК имеет структуру выполнения, близкую к алгоритмам (одинаковые точки входа и выхода, логика условных и безусловных переходов, вызовы других подпрограмм и т. п.), то утилиты создания по нему алгоритмов не являются технически сверхсложными. Вид алгоритмов может быть различным – как классическая блок-схема, так и структурограмма [13], Дракон-схемы [14] или даже собственный, специализированный для этого, псевдокод.

Формальная запись шага (Step 2.5), где *Algorithm* – алгоритм, полученный из псевдокода на шаге, имеет следующий вид:

$$Algorithm = Step_{2.5}(PseudoCode).$$

Так, с помощью шага, все множество псевдокода может быть преобразовано в его алгоритмы:

$$Algorithms = \{Algorithm_i\},$$

где *Algorithms* – множества алгоритмов псевдокода; *Algorithm_i* – *i*-й алгоритм из множества (для упрощения индекс *i* далее опустим).

Примером шага может быть применение различных генераторов блок-схем алгоритмов по коду [15], одним из которых является Visustin (на Интернет-ресурсе <https://www.aivosto.com/visustin.html>), поддерживающий ИК на 49 языках программирования. Естественно, ручное восстановление алгоритма экспертом также считается подходящим способом, имеющим свои достоинства и недостатки перед автоматическими программными.

В качестве условной альтернативы могут применяться IDA Pro или Ghidra, поскольку утилиты позволяют выводить граф потока управления подпрограммы, который тождественен ее алгоритму, но содержащий в элементах блоки МК вместо псевдокода. И хотя такой вид существенно менее понятен эксперту, однако он все же дает определенное представление об алгоритме подпрограммы.

Необходимо отметить, что процесс запутывания кода для понимания (т. н. обфускация) может применяться не только к МК, но и к ИК или же алгоритмам [16], что существенно усложняет задачу восстановления последних.

Шаг 2.6. Анализ алгоритмов подпрограмм для создания схемы внутрипрограммного взаимодействия

Шаг заключается в анализе всех алгоритмов одной Программы для построения схемы их обмена информационными объектами (значениями переменных, включая массивы и структуры) – т. е. внутри программного взаимодействия. Так, если один алгоритм вызывает другой (посредством соответствующей подпрограммы), имеющий аргументы и возвращающий значения, то можно говорить о полноценном двухстороннем внутрипрограммном взаимодействии. Отсутствие аргументов и возвращаемых значений у второго алгоритма соответствует вырожденному случаю взаимодействия посредством «пустых информационных объектов».

Формальная запись шага (*Step_{2.6}*) имеет следующий вид:

$$Scheme^{IntoProgramInteraction} = Step_{2.6}(Algorithms),$$

где *Scheme^{IntoProgramInteraction}* – схема, описывающая взаимодействие алгоритмов в Программе.

И хотя такая схема близка к графу вызовов подпрограмм, построение которого является достаточно тривиальной задачей для любого анализатора ИК и МК, однако полноценный учет всех входных и выходных параметров требует проведения дополнительного анализа. Такой граф вызовов, отображаемый в IDA Pro в окне WinGraph32, приведен на рисунке 4. Граф содержит взаимные вызовы

подпрограмм (с алгоритмами), но без указания передаваемых и возвращаемых значений; для этого требуется дополнительный анализ логики алгоритмов и точек их взаимного вызова.

Шаг 2.7. Анализ Программной системы для создания схемы межпрограммного взаимодействия

Шаг заключается в анализе всех Программ в Программной системе для построения схемы их обмена информационными объектами (параметрами вызовов, содержимым внешних файлов и т. п.) – т. е. межпрограммного взаимодействия [17]. Такой процесс был описан автором ранее в виде соответствующей модели [18], построенной на следующих 5 типах файлов: PE-, ELF-, JBC-, CIL- и скриптовых Программах; а также, на 3 типах взаимодействий: прямом вызове, прямом импорте и косвенном обмене. Так, если IDA Pro вызывает Программу WinGraph32 (в операционной системе Windows), то взаимодействие происходит между двумя PE-Программами путем прямого вызова; если же IDA Pro посредством выполнения пользовательских скриптов создает внешний файл, который затем считывается Python-скриптом, то взаимодействие происходит между PE- и скриптовой Программами путем косвенного обмена.

Формальная запись шага (*Step_{2.7}*) имеет следующий вид:

$$\{ Scheme^{InterProgramInteraction} = Step_{2.7}(Programs), \\ Programs \equiv \{ Program_i \} \},$$

где *Scheme^{InterProgramInteraction}* – схема, описывающая межпрограммное взаимодействие в Программной системе; *Programs* – множества Программ; *Program_i* – *i*-я Программа из множества (для упрощения индекс *i* далее опустим).

Ранее автором (с коллегами) была разработана утилита построения таких схем для PE-Программ и их прямого импорта. Пример визуализации схемы с помощью утилиты Gephi [19] для продукта Git в ОС Windows 10 приведен на рисунке 5. Анализ позволяет сделать вывод, что схема содержит основной центральный узел – для библиотеки kernel32.dll, ряд вторичных центров – для библиотек msvcrt.dll, user32.dll, advapi32.dll, kernel32.dll, а также обособленную группу точек – для библиотеки mscoree.dll (отмечено на рисунке пунктирным квадратом).

Шаг 2.8. Анализ схемы внутрипрограммного и межпрограммного взаимодействия для выявления интерфейсов

Шаг состоит из трех следующих действий. Во-первых, в анализе общей схемы обмена информационными объектами между Программами для определения связывающих их частных каналов; например, скрипт может вызывать PE-Программы с передачей им определенных аргументов командной строки.

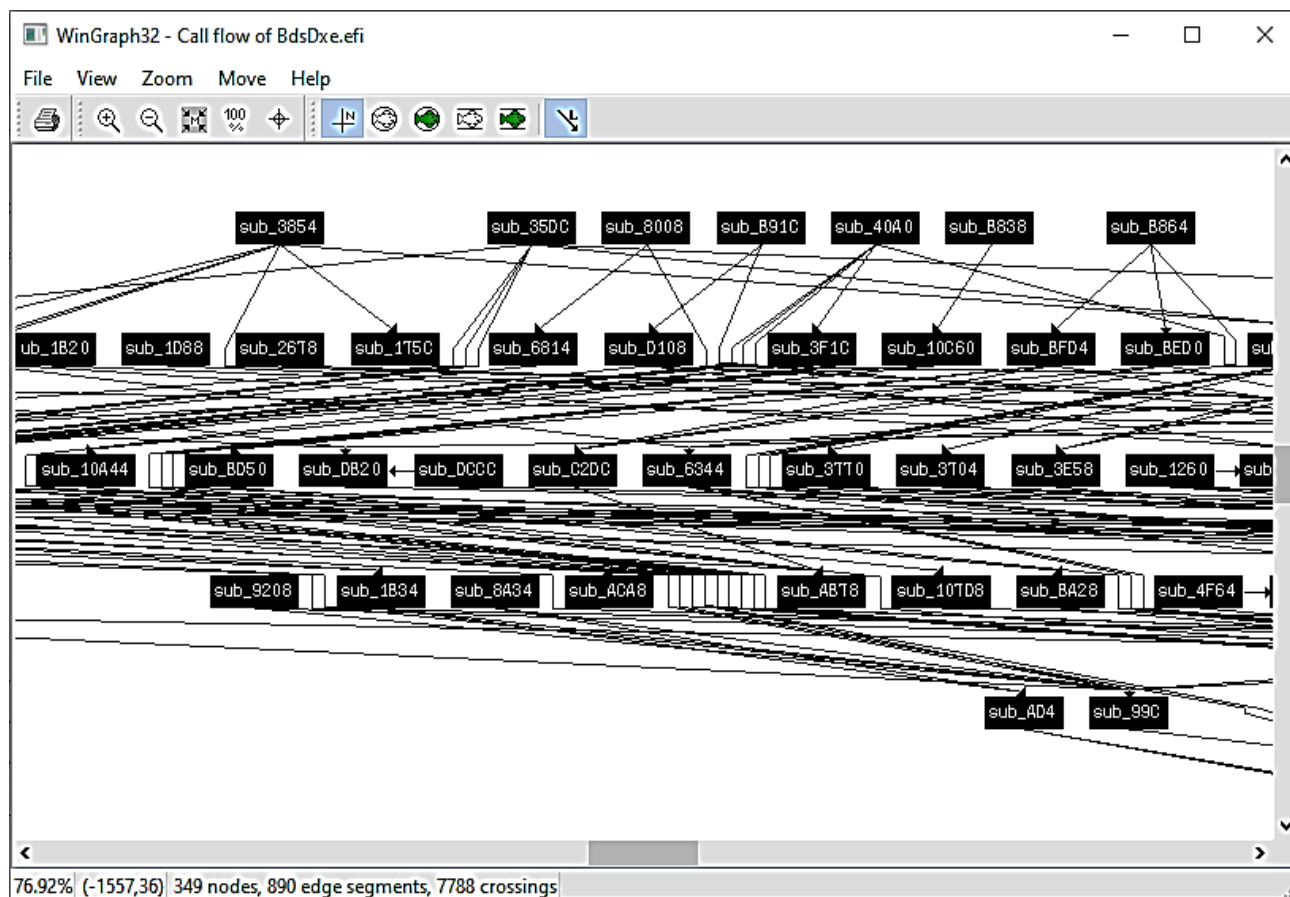


Рис. 4. Пример окна WinGraph32 в утилите IDA Pro

Fig. 4. WinGraph32 Window Example in the IDA Pro Utility

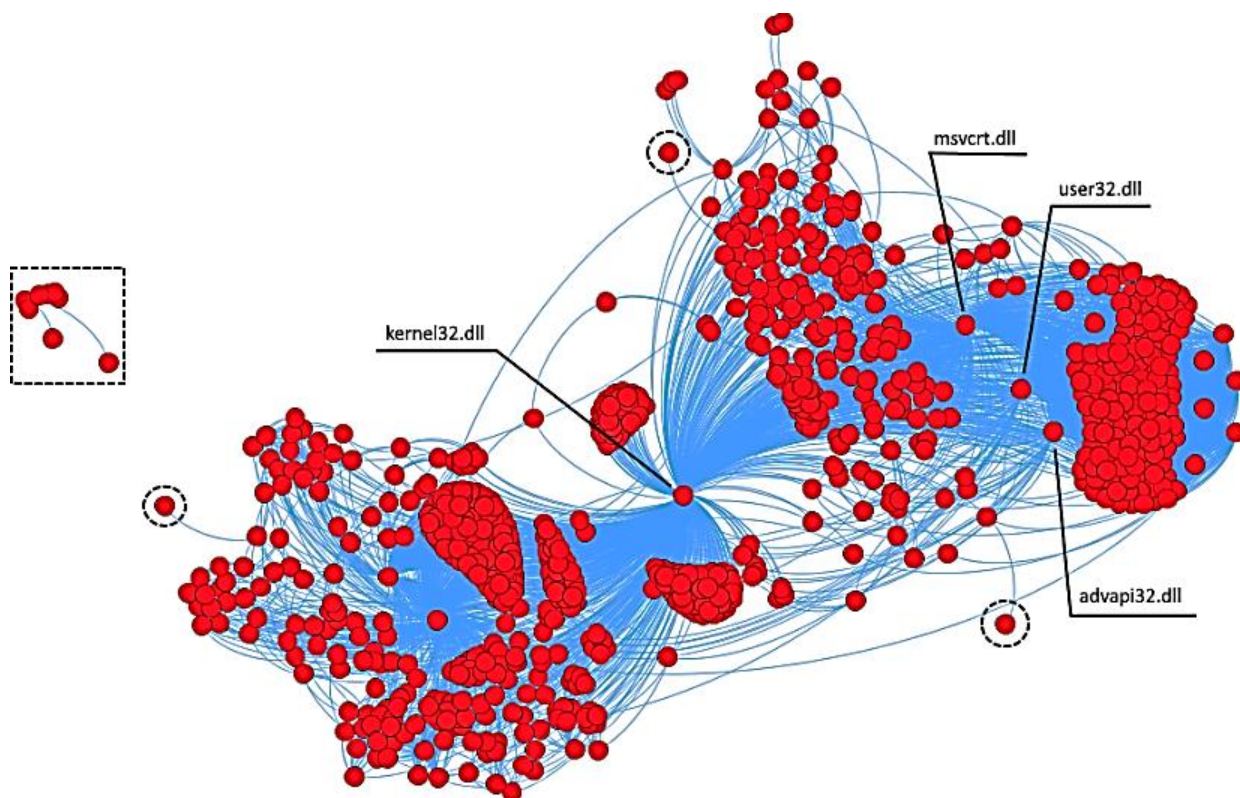


Рис. 5. Пример визуализации межпрограммного взаимодействия Программ в продукте Git

Fig. 5. Visualization of Inter-Program Interaction Example of Programs in the Git Product

Во-вторых, в анализе схемы обмена объектами между подпрограммами в рамках каждой Программы для определения ее входных и выходных частных каналов связи с окружением; например, Программа может анализировать аргументы командной строки и выдавать результат на консоль. И, в-третьих, в формировании набора обобщенных информационных каналов (из частных), которые бы связывали подпрограммы одной Программы с подпрограммами другой; такие каналы будем отождествлять с понятием интерфейсов, поскольку они в некотором смысле размывают границы между Программами и позволяют оперировать лишь множеством подпрограмм во всей Программной системе. Естественно, часть интерфейсов будет «смотреть» наружу из Программной системы, поскольку предназначены для взаимодействия с «внешним миром» – пользователями, сетью и т. д.

В классической Программе, выполняемой в операционной системе (далее – ОС), входной частью интерфейса является точка входа (указанная в за-

головках PE и ELF) – тот адрес МК, который начинает выполняться при запуске; данная подпрограмма получает на вход список аргументов командной строки при запуске, а на выходе возвращает статус завершения работы. В DXE-драйверах присутствует аналогичная подпрограмма, но принимающая на вход указатели на служебные объекты.

Также могут быть и неявные каналы взаимодействия с Программой, такие, как чтение или запись данных во внешний файл. Потенциальным признаком такой подпрограммы, связанной с интерфейсом, является ее терминальность в схеме внутрипрограммного взаимодействия – если она не вызывается другими подпрограммами, то в нее могут передаваться входные данные; аналогично, если она не вызывает другие подпрограммы, то через нее могут возвращаться данные из Программы.

Формальная запись шага ($Step_{2.8}$), где $Interfaces$ – множества интерфейсов Программ; $Interface_i$ – i -й интерфейс из множества (для упрощения индекс i далее опустим), имеет вид (2).

$$\left\{ \begin{array}{l} Interfaces = Step_{2.8}(Scheme^{IntoProgramInteraction}, Scheme^{InterProgramInteraction}) \\ Interfaces \equiv \{Interface_i\} \end{array} \right. \quad (2)$$

Примером шага может быть сбор информации о точках входа в Программу (указанных в поле `AddressOfEntryPoint` для PE-заголовка и `e_entry` для ELF-заголовка), поиска листьев в графовидной схеме внутрипрограммного взаимодействия, анализа кода подпрограмм на предмет наличия API-функций файловой системы (`open()` для открытия файла в стандартной библиотеке языка C) и работы с сетью (`socket()` для открытия сокета) и др.

Шаг 2.9. Анализ алгоритмов подпрограмм и интерфейсов для восстановления их общей архитектуры

Шаг заключается в анализе алгоритмов для понимания их задача-ориентированности, что позволит объединить близкие с этой позиции подпрограммы в логические единицы архитектуры – модули; связи же между модулями будут определяться внутренним взаимодействием подпрограмм. Дальнейшее же объединение модулей в более иерархические высокоуровневые сущности сформирует остальные логические слои архитектуры вплоть до требуемого уровня абстракции.

Анализ интерфейсов между Программами (если их несколько) позволит соединить модули, полученные из разных Программ, в единую архитектуру, отражающую уже не отдельную Программу, а их целую Программную систему. Графическая иллюстрация такого способа взаимодействия подпрограмм в Программной системе – на рисунке 6.

Для наглядности представления архитектуры предположим, что в Программе типа «арифметический калькулятор» были выделены и интерпретированы 8 подпрограмм со следующими задачами: сложение, умножение, вычитание и деление чисел; ввод чисел; вывод результата; логирование работы; сообщение об ошибке.

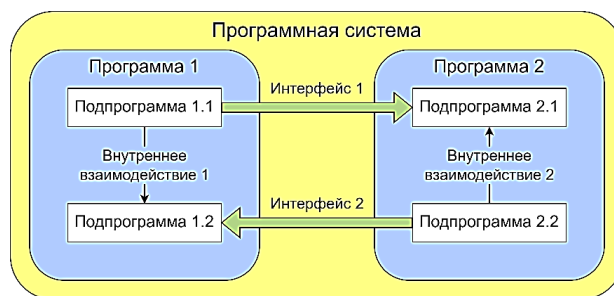


Рис. 6. Графическая иллюстрация способа взаимодействия подпрограмм в Программной системе

Fig. 6. Graphic Illustration of the Interaction Subroutines Method in the Software System

Тогда на данном шаге первые 4 подпрограммы могут быть объединены в модуль «Операции», подпрограммы 5 и 6 – в модуль «Взаимодействие с пользователем»; а 7 и 8 – в модуль «Системный функционал». Затем первые 2 модуля можно объединить в более высокоуровневый модуль – «Основной функционал», построив таким образом иерархическую архитектуру. При этом в архитектуре будут отмечены входной и выходной интер-

фейсы к пользователю – в модуле с подпрограммами 5 и 6, а также выходной интерфейс к ОС – в модуле с подпрограммами 7 и 8. Необходимо заметить, что приведенный способ записи архитектуры является одним из возможных [20], поскольку существуют более сложные представления, например, с применением UML; однако, здесь и далее будем рассматривать именно такой простой вид архитектуры.

Формальная запись шага ($Step_{2.9}$), состоящая из двух частей – получение частной архитектуры каждой Программы ($Step_{2.9.1}$) и общей архитектуры всей Программной системы ($Step_{2.9.2}$), имеет следующий вид:

$$\left\{ \begin{array}{l} Architecture = Step_{2.9.1}(Subroutine) \\ Architectures \equiv \{Architecture\} \\ \overline{Architecture} = Step_{2.9.2}(Architectures, Interfaces) \end{array} \right.,$$

где *Architecture* – частная архитектура МК отдельной Программы, полученная на шаге; *Architectures* – множества архитектур для всех Программ; $\overline{Architecture}$ – общая архитектура всей Программной системы, полученная на шаге.

Примером шага может быть построение одноуровневой архитектуры для DXE-драйвера *BdsDxe.efi* в виде совокупности 3-х модулей: «BDS-фаза» – реализующего логику одноименной UEFI фазы, «Поддержка языка» – поддерживающего настройки языка интерфейса, и «Запись аппаратных ошибок» – устанавливающего уровень поддержки записей об аппаратных ошибках. Соответственно, точка входа в Программу расположена в 1-м модуле, который вызывает оставшиеся 2.

Необходимо отметить, что в случае наличия лишь одной Программы интерфейсы между Программами будут отсутствовать, а следовательно, частная и общая архитектуры окажутся тождественны: $Architecture \equiv \overline{Architecture}$.

Шаг 2.10. Анализ архитектуры для восстановления ее концептуальной модели

Шаг заключается в анализе модулей, их связей и интерфейсов (т. е. архитектуры) Программной си-

стемы для восстановления ее концептуальной модели, под которой понимается совокупность взаимосвязанных основополагающих сущностей, представляющая «смысл» всей системы с высокоуровневой позиции. В принципе, концептуальная модель близка к онтологической, хотя и предназначена для отражения специфики конкретной Программы, а не всей предметной области.

Формально запись текущего шага ($Step_{2.10}$), где $Model^{Conceptual}$ – концептуальная модель Программы, имеет следующий вид:

$$Model^{Conceptual} = Step_{2.10}(\overline{Architecture}).$$

Примером шага для архитектуры *BdsDxe.efi* может быть обобщение модулей архитектуры и получение совокупности следующих элементов концепции – «Человек», «BDS-фаза» и «Настройки модуля»; при этом 1-й элемент связан со 2-м, который связан с 3-м. По аналогии с архитектурой, в случае наличия одной Программы концептуальная модель, будет относиться только к ней.

Шаг 2.11. Анализ восстановленной Метаинформации для описания функционала МК

Шаг заключается в составлении итогового функционала МК [21] на основании анализа Метаинформации различного типа, восстановленной в результате предыдущих шагов – псевдокода после 2.4, алгоритмов после 2.5, архитектуры после 2.9 и концептуальной модели после 2.10. Таким образом, будет получен основной результат исследования, при этом касающийся не только данного этапа, а всего РИ, поскольку последующие этапы носят уточняющий и документирующий характер. В простейшем случае описание функционала может быть перечислением внутренних действий Программ (по алгоритмам их подпрограмм), некоторых деталей их реализации (по псевдокоду), а также решаемых задач (по архитектуре), которые сформулированы на едином базисе сущностей (по концептуальной модели).

Формальная запись шага ($Step_{2.11}$), где *Metainfo* – все Метаинформация, восстановленная из МК Программ на шаге, имеет вид (3).

$$\left\{ \begin{array}{l} Functionality = Step_{2.11}(Metainfo) \\ Metainfo \equiv \langle PseudoCodes, Algorithms, \overline{Architecture}, Model^{Conceptual} \rangle \end{array} \right. \quad (3)$$

Примером шага для DXE-драйвера *BdsDxe.efi* может быть анализ архитектуры из связанных с задачами модулей – «BDS-фаза», «Поддержка языка», «Запись аппаратных ошибок», которые согласно концептуальной модели обеспечивают взаимодействие пользователя с BDS-фазой UEFI, работающей согласно настройкам (из NVRAM); а все детали решения этих задач могут быть сформулированы на основании логики работы конкретных подпрограмм и

их псевдокода. В результате будет получено человеко-ориентированное описание драйвера, хорошо понятное эксперту.

Шаг 2.12. Поиск Уязвимостей по восстановленной Метаинформации

Шаг заключается в анализе восстановленной Метаинформации на предмет наличия в ней Уязвимостей, поскольку одним из целевых назначений РИ

является проверка Программы на предмет ее безопасности. Так, каждый тип Метаинформации связан со своим структурным уровнем Уязвимости [22]: псевдокод – с низким, алгоритм – со средним, архитектура – с высоким и концептуальная модель – со сверхвысоким или концептуальным. И хотя для поиска Уязвимостей уже существует ряд программных средств, однако их наибольшая результативность оказывается для низкоуровневых и частично среднеуровневых Уязвимостей; для поиска остальных классов Уязвимостей, как правило, требуется привлечение экспертов по безопасности программного кода.

Формальная запись шага ($Step_{2.12}$) имеет следующий вид:

$$\left\{ \begin{array}{l} Vulnerabilities = Step_{2.12}(Metainfo) \\ Vulnerabilities \equiv \{Vulnerability_i\} \\ Class_{Vulnerability} \in \left\{ \begin{array}{l} Vulnerability^{LowLevel}, \\ Vulnerability^{MediumLevel}, \\ Vulnerability^{HighLevel}, \\ Vulnerability^{ConceptualLevel} \end{array} \right\} \end{array} \right\},$$

где $Vulnerabilities$ – множества Уязвимостей, обнаруженных на шаге; $Vulnerability_i$ – i -я Уязвимость из множества (для упрощения, индекс i далее опустим); $Class_{Vulnerability}$ – класс, связанный с Уязвимостями согласно их структурного уровня:

- низкоуровневой ($Vulnerability^{LowLevel}$);
- среднеуровневой ($Vulnerability^{MediumLevel}$);
- высокоуровневой ($Vulnerability^{HighLevel}$);
- концептуальной ($Vulnerability^{ConceptualLevel}$).

Примером шага для $BdsDxe.efi$ в виде Уязвимостей, которые обнаруживаемы по различной Метаинформации, может быть следующий. Применение переменных меньшего размера, чем возможные значения, в псевдокоде могут привести к хранению неверных настроек BDS-фазы. Ошибки в логике выбора наилучшего раздела для загрузки ОС после BDS фазы могут привести к невозможности запуска всей системы. Отсутствие связи модуля BDS-фазы с

настройками будет означать невозможность управлять логикой ее работы. Если в Образ материнской платы внедрен модуль доверенной загрузки, выполненный в виде DXE-драйвера [23], то отсутствие в концептуальной модели связи DXE-драйвера $BdsDxe.efi$ (который как раз отвечает за выбор загрузочного устройства) с этим модулем скорее всего будет означать небезопасность загрузки ОС, поскольку отсутствует механизм управления ею.

Шаг 2.13. Применение машинного обучения к подпрограммам для дополнительного восстановления Метаинформации

Шаг является достаточно технически сложным и мало затрагиваемым в современных научных публикациях, поскольку заключается в применении моделей и методов машинного обучения для восстановления псевдокода из МК подпрограмм (а также глобальных данных) – т. е. является некой «интеллектуальной» декомпиляцией [24]. Таким образом, уже полученный псевдокод (на Шаг 2.4) может быть дополнен на данном шаге новыми сведениями. Шаг может реализовываться путем классификации различных последовательностей инструкций МК по соответствующим конструкциям в языках программирования среднего и высокого уровня. Также в интересах этого возможно применение авторского метода генетической декомпиляции [25], основанного на использовании генетических алгоритмов для оптимального подбора конструкций исходного кода, в точности компилируемых в исследуемый МК.

Формальная запись шага ($Step_{2.13}$), в которой $PseudoCode'$ – псевдокод подпрограммы, уточненный в процессе статического исследования на шаге; $MachineLearning$ – применяемые для уточнения модели и методы машинного обучения; « $\rightarrow$ » – операция изменения сущности (в данном случае обновление Метаинформации); « $\cup$ » – объединение сущностей (в данном случае Метаинформации и уточнений в псевдокоде), имеет вид (4).

$$\left\{ \begin{array}{l} PseudoCode' = Step_{2.13}(Subroutine, GlobalDatas, MachineLearning) \\ Metainfo \rightarrow Metainfo \cup PseudoCode' \end{array} \right. \quad (4)$$

Примером шага для подпрограммы $sub_1D88()$ из Шага 2.4 может быть подбор с применением генетического алгоритма конструкций единственной строки ее псевдокода таким образом, чтобы она компилировалась в 5 инструкций ее МК.

Шаг 2.14. Применение машинного обучения к восстановленной Метаинформации для ее уточнения

Как и Шаг 2.14, данный шаг является также мало изученным, хотя для восстановления Метаинформации ряд задач, решаемых с помощью машинного

обучения, подходят достаточно хорошо. Поскольку основная часть Метаинформации на данный момент РИ уже была восстановлена (в основном ручным способом, хотя и с применением автоматизирующих средств), то шаг носит уточняющий характер – т. е. дополняет или корректирует их, например, следующим образом. Во-первых, типовые наборы конструкций псевдокода могут быть отнесены к более абстрактным конструкциям алгоритмов. Во-вторых, подобная классификация в купе с кластеризацией позволяет выделять близкие подпрограммы (по их алгоритмам) в отдельные модули и их высокоуров-

невые группы [26]. И, в-третьих, аналогичным образом можно получить концептуальную модель Программной системы (или отдельной Программы) из ее архитектуры, определив категории всех верхних модулей и образовав их логические связи.

Формальная запись текущего шага ($Step_{2.14}$), где *Metainfo'* – Метаинформация, уточненная в процессе статического исследования на шаге; *MachineLearning* – применяемые для уточнения модели и методы машинного обучения, имеет следующий вид:

$$\{Metainfo' = Step_{2.14}(Metainfo, MachineLearning) \\ Metainfo \rightarrow Metainfo \cup Metainfo'\}$$

Пример шага для DXE-драйвера *BdsDxe.efi* может заключаться в следующем. Во-первых, псевдокод для подпрограммы *sub_1D88()*, состоящий из вызова другой подпрограммы *sub_6A8C()* с двумя параметрами, может быть с помощью машинного обучения проклассифицирован как следующая конструкция блок-схемы алгоритма – «вызов предопределенного процесса (функции) с передачей параметров». Во-вторых, множество подпрограмм, вызывающих друг друга и расположенных в близком диапазоне адресов, могут быть отнесены к одному модулю архитектуры Программы (как частной архитектуре Программной системы); вызовы же подпрограмм между различными диапазонами

будут означать связи между модулями, а их названия (или описания) могут быть сформированы из совокупности строковых констант в МК этих модулей (например, с помощью моделей Word2Vec [27]). И, в-третьих, сущности концептуальной модели их взаимосвязи могут быть получены путем категорирования названия (или описания) модулей архитектуры с учетом их иерархии.

Заключение

В данной части цикла статей был описан 2-й этап методологии реверс-инжиниринга, являющийся наиболее технически сложным. Для этого, в частности, сделан обзор существующих методов и средств статического анализа машинного кода.

Новизна соответствует указанной в первой части цикла и заключается в системности и масштабности охвата решения задачи реверс-инжиниринга. Теоретическая и практическая значимость аналогична, но в отличие от первой части, посвященной подготовке МК, относится к его статическому исследованию.

Продолжением исследования должно стать описание оставшихся 3-го и 4-го этапов методологии реверс-инжиниринга, а также представление итоговой схемы процесса и сведение шагов всех этапов в единую систему.

Окончание следует...

Список источников

1. Израйлов К.Е. Методология проведения реверс-инжиниринга машинного кода. Часть 1. Подготовка объекта исследования // Труды учебных заведений связи. 2023. Т. 9. № 5. С. 79–90. DOI:10.31854/1813-324X-2023-9-5-79-90
2. Падарян В.А., Гетьман А.И., Соловьев М.А., Бакулин М.Г., Борзилов А.И., Каушан В.В. Методы и программные средства, поддерживающие комбинированный анализ бинарного кода // Труды Института системного программирования РАН. 2014. Т. 26. № 1. С. 251–276.
3. Бугеря А.Б., Ефимов В.Ю., Кулагин И.И., Падарян В.А., Соловьев М.А., Тихонов А.Ю. Программный комплекс для выявления недекларированных возможностей в условиях отсутствия исходного кода // Труды Института системного программирования РАН. 2019. Т. 31. № 6. С. 33–64. DOI:10.15514/ISPRAS-2019-31(6)-3
4. Долгова К.Н., Чернов А.В., Деревенец Е.О. Методы и алгоритмы восстановления программ на языке ассемблера в программы на языке высокого уровня // Проблемы информационной безопасности. Компьютерные системы. 2008. № 3. С. 54–68.
5. Новиков В.А., Ломако А.Г., Еремеев М.А., Петренко А.С. Выявление и нейтрализация недекларированных возможностей программ // Proceedings of the 2017 Symposium on Cybersecurity of the Digital Economy (CDE'17, Иннополис, Россия, 19–20 сентября 2017). Санкт-Петербург: Издательский Дом «Афина», 2017. С. 284–287.
6. Ревнивых А.В., Велижанин А.С. Методика автоматизированного формирования структуры дизассемблированного листинга // Кибернетика и программирование. 2019. № 2. С. 1–16. DOI:10.25136/2306-4196.2019.2.28272
7. Bhardwaj V., Kukreja V., Sharma C., Kansal I., Popali R. Reverse Engineering-A Method for Analyzing Malicious Code Behavior // Proceedings of the International Conference on Advances in Computing, Communication, and Control (ICAC3, Mumbai, India, 03–04 December 2021). IEEE, 2022. PP. 1–5. DOI:10.1109/ICAC353642.2021.9697150
8. Черчесов А.Э. Фазы загрузки UEFI и способы контроля исполняемых образов // Вопросы защиты информации. 2018. № 2(121). С. 51–53.
9. Zhang D., Zhang Z., Jiang B., Tse T.H. The Impact of Lightweight Disassembler on Malware Detection: An Empirical Study // Proceedings of the 42nd Annual Computer Software and Applications Conference (Tokyo, Japan, 23–27 July 2018). IEEE, 2018. PP. 620–629, DOI:10.1109/COMPSAC.2018.00094
10. David A.P. Ghidra Software Reverse Engineering for Beginners: Analyze, identify, and avoid malicious code and potential threats in your networks and systems. UK: Packt Publishing Ltd, 2021. 322 p.
11. Буйневич М.В., Израйлов К.Е. Автоматизированное средство алгоритмизации машинного кода телекоммуникационных устройств // Телекоммуникации. 2013. № 6. С. 2–9.
12. Буйневич М.В., Израйлов К.Е. Метод алгоритмизации машинного кода телекоммуникационных устройств // Телекоммуникации. 2012. № 12. С. 2–6.

13. Селиверстова И.А. Разработка программного обеспечения построения XML описания кода // Современные научные исследования и инновации. 2016. № 2(58). С. 102–104.
14. Митькин С.Б. Автоматное программирование на языке Дракон // Программная инженерия. 2019. Т. 10. № 1. С. 3–13. DOI:10.17587/prin.10.3-13
15. Вохмин А.А., Евдокимова О.А., Малявко А.А. Визуально-графическая система программирования на основе разработки блок-схем алгоритмов. Конвертирование текстов программ на различных языках программирования в блок-схемы и обратно // Южно-Сибирский научный вестник. 2021. № 3(37). С. 49–57. DOI:10.25699/SSSB.2021.37.3.013
16. Pakonen A. Obfuscation of function block diagrams // Proceedings of the 28th International Conference on Emerging Technologies and Factory Automation (ETFA, Sinaia, Romania, 12–15 September 2023). IEEE, 2023. PP. 1–7. DOI:10.1109/ETFA54631.2023.10275363
17. Ипатов П.С. Технологии межпрограммных интерфейсов // Science Time. 2016. № 9(33). С. 115–118.
18. Буйневич М.В., Ганов Г.А., Израйлов К.Е. Интеллектуальный метод визуализации взаимодействий программ в интересах аудита информационной безопасности операционной системы // Информатизация и связь. 2020. № 4. С. 67–74.
19. Yang J., Cheng C., Shen S., Yang S. Comparison of complex network analysis software: Citespace, SCI2 and Gephi // Proceedings of the 2nd International Conference on Big Data Analysis (Beijing, China, 10–12 March 2017). IEEE, 2017. PP. 169–172. DOI:10.1109/ICBDA.2017.8078800
20. Gardazi S.U., Shahid A.A. Survey of software architecture description and usage in software industry of Pakistan // Proceedings of the International Conference on Emerging Technologies (Islamabad, Pakistan, 19–20 October 2009). IEEE, 2009. PP. 395–402. DOI:10.1109/ICET.2009.5353137
21. Sharma K.I., Dubey S.K., Gaurav P., Prachi. Functionality Assessment of Software System using Fuzzy Approach // Proceedings of the 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO, Noida, India, 04–05 June 2020). IEEE, 2020. PP. 1206–1209. DOI:10.1109/ICRITO48877.2020.9197795
22. Kotenko I., Izrailov K., Buinevich M., Saenko I., Shorey R. Modeling the Development of Energy Network Software, Taking into Account the Detection and Elimination of Vulnerabilities // Energies. 2023. Vol. 16. Iss. 13. P. 5111. DOI:10.3390/en16135111
23. Израйлов К.Е., Покусов В.В. Создание программной объектно-ориентированной платформы для разработки UEFI модулей // X международная научно-техническая и научно-методическая конференция «Актуальные проблемы инфотелекоммуникаций в науке и образовании» (АПИНО 2021, Санкт-Петербург, Россия, 24–25 февраля 2021). Санкт-Петербург: СПбГУТ. 2021. Т. 2. С. 246–250.
24. Yu S.-Y., Achamyeh Y.G., Wang C., Kocheturov A., Eisen P., Al Faruque M.A. CFG2VEC: Hierarchical Graph Neural Network for Cross-Architectural Software Reverse Engineering // Proceedings of the 45th International Conference on Software Engineering: Software Engineering in Practice (Melbourne, Australia, 14–20 May 2023). IEEE, 2023. PP. 281–291. DOI:10.1109/ICSE-SEIP58684.2023.00031
25. Израйлов К.Е. Концепция генетической декомпиляции машинного кода телекоммуникационных устройств // Труды учебных заведений связи. 2021. Т. 7. № 4. С. 95–109. DOI:10.31854/1813-324X-2021-7-4-95-109
26. Израйлов К.Е., Умаралиев И.В. Гипотетический метод восстановления модулей архитектуры машинного кода с целью выявления высокоуровневых уязвимостей // XII международная научно-техническая и научно-методическая конференция Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2023, Санкт-Петербург, Россия, 28 февраля – 01 марта 2023). Санкт-Петербург: СПбГУТ, 2023. Т. 1. С. 577–581.
27. Wang R., Shi Y. Research on application of article recommendation algorithm based on Word2Vec and TfIdf // The Proceedings of International Conference on Electrical Engineering, Big Data and Algorithms (Changchun, China, 25–27 February 2022). IEEE, 2022. PP. 454–457. DOI:10.1109/EEBDA53927.2022.97448244

References

1. Izrailov K. Methodology for Machine Code Reverse Engineering. Part 1. Preparation of the Research Object. *Proceedings of the Telecommun. Univ.* 2023;9(5):79–90. DOI:10.31854/1813-324X-2023-9-5-79-90
2. Padaryan V.A., Getman A.I., Solovov M.A., Bakulin M.G., Borzilov A.I., Kaushan V.V. Methods and software tools supporting combined binary code analysis. *Proceedings of ISP RAS.* 2014;26(1):251–276.
3. Bugerya A.B., Yefimov V.Yu., Kulagin I.I., Padaryan V.A., Solovov M.A., Tikhonov A.Yu. Program complex for detecting undeclared capabilities in the absence of source code. *Proceedings of ISP RAS.* 2019;31(6):33–64. DOI:10.15514/ISPRAS-2019-31(6)-3
4. Dolgova K.N., Chernov A.V., Derevenets Ye.O. Methods and algorithms for restoring assembly language programs into high-level language programs. *Information Security Problems. Computer Systems.* 2008;3:54–68.
5. Novikov V.A., Lomako A.G., Yermeev M.A., Petrenko A.S. Identification and neutralization of undeclared program features. *Proceedings of the 2017 Symposium on Cybersecurity of the Digital Economy, CDE'17, 19–20 September 2017, Innopolis, Russia.* St. Petersburg: Afina Publ.; 2017. p.284–287.
6. Revnivikh A.V., Velizhanin A.S. Automated Formation Methodology of disassembled listing structure. *Cybernetics and Programming.* 2019;2:1–16. DOI:10.25136/2306-4196.2019.2.28272
7. Bhardwaj V., Kukreja V., Sharma C., Kansal I., Popali R. Reverse Engineering-A Method for Analyzing Malicious Code Behavior. *Proceedings of the International Conference on Advances in Computing, Communication, and Control, ICAC3, 03–04 December 2021, Mumbai, India.* IEEE; 2022. p.1–5. DOI:10.1109/ICAC353642.2021.9697150
8. Chershevov A.E. UEFI boot phases and how to control executable images. *Voprosy zashchity informatsii.* 2018;2(121): 51–53.

9. Zhang D., Zhang Z., Jiang B., Tse T.H. The Impact of Lightweight Disassembler on Malware Detection: An Empirical Study. *Proceedings of the 42nd Annual Computer Software and Applications Conference, 23–27 July 2018, Tokyo, Japan*. IEEE; 2018. p.620–629. DOI:10.1109/COMPSAC.2018.00094
10. David A.P. *Ghidra Software Reverse Engineering for Beginners: Analyze, identify, and avoid malicious code and potential threats in your networks and systems*. Packt Publishing Ltd; 2021. 322 p.
11. Buinevich M.V., Izrailov K.E. Automated tool for machine code algorithmization of telecommunication devices. *Telekommunikatsii*. 2013;6:2–9.
12. Buinevich M.V., Izrailov K.E. Algorithmization method for machine code of telecommunication devices. *Telekommunikatsii*. 2012;12:2–6.
13. Seliverstova I.A. Development of software for building XML code description. *Modern scientific researches and innovations*. 2016;2(58):102–104.
14. Mitkin S.B. Automata programming in the Dragon language. *Software Engineering*. 2019;10(1):3–13. DOI:10.17587/prin.10.3-13
15. Vokhmin A.A., Yevdokimova O.A., Malyavko A.A. Visual-graphical programming system based on the development of algorithm flowcharts. Converting program texts in different programming languages into flowcharts and back again. *South-Siberian Scientific Bulletin*. 2021;3(37):49–57. DOI:10.25699/SSSB.2021.37.3.013
16. Pakonen A. Obfuscation of function block diagrams. *Proceedings of the 28th International Conference on Emerging Technologies and Factory Automation, ETFA, 12–15 September 2023, Sinaia, Romania*. IEEE; 2023. p.1–7. DOI:10.1109/ETFA54631.2023.10275363
17. Ipatov P.S. Technologies of interprogram interfaces. *Science Time*. 2016;9(33):115–118.
18. Buinevich M.V., Ganov G.A., Izrailov K.E. An intelligent method for visualizing program interactions in the interest of operating system information security auditing. *Informatization and communication*. 2020;4:67–74.
19. Yang J., Cheng C., Shen S., Yang S. Comparison of complex network analysis software: Citespace, SCI2 and Gephi. *Proceedings of the 2nd International Conference on Big Data Analysis Beijing, 10–12 March 2017, Beijing, China*. IEEE; 2017. p.169–172. DOI:10.1109/ICBDA.2017.8078800
20. Gardazi S.U., Shahid A.A. Survey of software architecture description and usage in software industry of Pakistan. *Proceedings of the International Conference on Emerging Technologies, 19–20 October 2009, Islamabad, Pakistan*. IEEE; 2009. p.395–402. DOI:10.1109/ICET.2009.5353137
21. Sharma K., Dubey S.K., Gaurav P., Prachi Functionality Assessment of Software System using Fuzzy Approach. *Proceedings of the 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions), ICRITO, 04–05 June 2020, Noida, India*. IEEE; 2020. p.1206–1209. DOI:10.1109/ICRITO48877.2020.9197795
22. Kotenko I., Izrailov K., Buinevich M., Saenko I., Shorey R. Modeling the Development of Energy Network Software, Taking into Account the Detection and Elimination of Vulnerabilities. *Energies*. 2023;16(13):5111. DOI:10.3390/en16135111
23. Izrailov K.E., Pokusov V.V. Creation of software object-oriented platform for UEFI modules development. *Proceedings of the X International Scientific-Technical and Scientific-Methodical Conference on Actual Problems of Infotelecommunications in Science and Education, 24–25 February 2021, St. Petersburg, Russia*. St. Petersburg: The Bonch-Bruевич Saint-Petersburg State University of Telecommunications Publ.; 2021. vol.2. p.246–250.
24. Yu S.-Y., Achamyel Y.G., Wang C., Kocheturov A., Eisen P., Al Faruque M.A. CFG2VEC: Hierarchical Graph Neural Network for Cross-Architectural Software Reverse Engineering. *Proceedings of the 45th International Conference on Software Engineering: Software Engineering in Practice, 14–20 May 2023, Melbourne, Australia*. IEEE; 2023. p.281–291. DOI:10.1109/ICSE-SEIP58684.2023.00031
25. Izrailov K.E. The concept of genetic decompilation machine code telecommunication devices. *Proceedings of the Telecommun. Univ*. 2021;7(4):95–109. DOI:10.31854/1813-324X-2021-7-4-95-109
26. Izrailov K.E., Umaraliev I.V. Hypothetical method of restoring machine code architecture modules in order to detect high-level vulnerabilities. *Proceedings of the XII International Scientific-Technical and Scientific-Methodical Conference on Actual Problems of Infotelecommunications in Science and Education, 28 February – 01 March 2023, St. Petersburg, Russia*. St. Petersburg: The Bonch-Bruевич Saint-Petersburg State University of Telecommunications Publ.; 2023. vol.1. p.577–581.
27. Wang R., Shi Y. Research on application of article recommendation algorithm based on Word2Vec and TfIdf. *Proceedings of the International Conference on Electrical Engineering, Big Data and Algorithms, 25–27 February 2022, Changchun, China*. IEEE; 2022. p.454–457. DOI:10.1109/EEBDA53927.2022.9744824

Статья поступила в редакцию 25.10.2023; одобрена после рецензирования 21.11.2023; принята к публикации 22.11.2023.

The article was submitted 25.10.2023; approved after reviewing 21.11.2023; accepted for publication 22.11.2023.

Информация об авторе:

**ИЗРАИЛОВ
Константин Евгеньевич**

кандидат технических наук, доцент, старший научный сотрудник лаборатории проблем компьютерной безопасности Санкт-Петербургского Федерального исследовательского центра Российской академии наук

 <https://orcid.org/0000-0002-9412-5693>

Обзорная статья

УДК 004.056; 004.8

DOI:10.31854/1813-324X-2023-9-6-83-94



Машинное обучение vs поиск уязвимостей в программном обеспечении: анализ применимости и синтез концептуальной системы

Николай Викторович Леонов¹, Leonov-nv@yandex.ru

Михаил Викторович Буйневич²✉, bmv1958@yandex.ru

¹Государственный научно-исследовательский институт прикладных проблем,
Санкт-Петербург, 191167, Российская Федерация

²Санкт-Петербургский университет государственной противопожарной службы МЧС России,
Санкт-Петербург, 196105, Российская Федерация

Аннотация: Работа посвящена проблеме поиска уязвимостей в программном обеспечении, а также возможностям применения такого перспективного направления информационных технологий, как машинное обучение. Для этого произведен обзор научных публикаций предметной области из российской и зарубежной базы цитирования. Осуществлен сравнительный анализ результатов обзора по следующим критериям: год публикации, область применения, идея, решаемая задача машинного обучения, степень реализации его моделей и методов; по каждому критерию сделаны основополагающие выводы. В итоге предложены 7 принципов построения новой концептуальной системы поиска уязвимостей в программном обеспечении с помощью машинного обучения, краткий смысл которых состоит в следующем: многостороннее исследование программы, комбинирование известных методов, использование машинного обучения в каждом методе и алгоритме управления ими, возможность корректировки работы экспертом, хранение информации в базе данных и ее синхронизации с внешними, рекомендательный характер относительно найденных уязвимостей; использование единой программно-аппаратной платформы. На основании декларируемых принципов разработана графическая схема такой системы.

Ключевые слова: информационная безопасность, поиск уязвимостей, машинное обучение, обзор, принципы, система поиска

Ссылка для цитирования: Леонов Н.В., Буйневич М.В. Машинное обучение vs поиск уязвимостей в программном обеспечении: анализ применимости и синтез концептуальной системы // Труды учебных заведений связи. 2023. Т. 9. № 6. С. 83–94. DOI:10.31854/1813-324X-2023-9-6-83-94

Machine Learning vs Software Vulnerability Detection: Applicability Analysis and Conceptual System Synthesis

Nikolay Leonov¹, Leonov-nv@yandex.ru

Mikhail Buinevich²✉, bmv1958@yandex.ru

¹State Research Institute of Applied Problems,
St. Petersburg, 191167, Russian Federation

²Saint-Petersburg University of State Fire Service of EMERCOM of Russia,
St. Petersburg, 196105, Russian Federation

Abstract: The article is devoted to the searching for vulnerabilities in software problem, as well as the possibilities of application of such a promising area in information technology as machine learning. For this purpose, a review of scientific publications in this area from Russian and foreign citation databases is made. A comparative analysis of the review's results was made according to the following criteria: publication year, application field, idea, solved problem of machine learning, degree of realization of its models and methods; for each criterion basic conclusions were drawn. As a result, 7 principles of building a new conceptual system of searching for vulnerabilities in software with the help of machine learning are proposed, the short meaning of which is as follows: program's multilateral study, combination of known methods, the use of machine learning in each method and algorithm of its management, the possibility of correcting the expert's work, storing information in a database and its synchronization with external, advisory nature of the found vulnerabilities; single software application usage. Based on the stated principles, a graphical scheme of such a system has been developed.

Keywords: information security, vulnerability search, machine learning, review, principles, search engine

For citation: Leonov N., Buinevich M. Machine Learning vs Software Vulnerability Detection: Applicability Analysis and Conceptual System Synthesis. *Proceedings of Telecommun. Univ.* 2023;9(6):83–94. DOI:10.31854/1813-324X-2023-9-6-83-94

Введение

Безопасность программного обеспечения (ПО) является актуальнейшей проблемой современного мира, практически полностью функционирующего на базе и законах информационных технологий. Одним из наиболее эффективных способов повышения безопасности ПО (хотя и не единственным) небезосновательно признан поиск уязвимостей в программах с последующей их нейтрализацией или попросту отказом от дальнейшего использования. Наличие уязвимостей в программах может приводить к критическим последствиям в любой области – начиная от бытовых «умных» устройств или персональных компьютеров и заканчивая обороноспособностью целой страны. Постоянно растущее количество такого рода информационных угроз позволяет утверждать, что данный способ (имеется в виду – поиск уязвимостей) достиг некоторого «научно-технического тупика», для выхода из которого требуется применение качественно новых подходов.

В качестве одного из них гипотетически может выступить машинное обучение (МО), уже доказавшее свою применимость для решения огромного спектра разнородных прикладных задач: программный инжиниринг [1], управление состоянием дорожного покрытия [2], построение сетей 5G [3], повышение обучаемости студентов [4], обеспечение непрерывности функционирования Интернета вещей [5], – и многие другие. Исходя из этих соображений, далее в статье будет проведен аналитический обзор возможностей МО, которые могут быть применены для поиска уязвимостей в ПО – что и определяет предметную область настоящего исследования.

Обзор релевантных работ

Произведем обзор научных публикаций отечественных и зарубежных ученых, посвященных задаче поиска уязвимостей в ПО, для решения которой применяются модели и методы МО; поиск

осуществлялся по базам РИНЦ и IEEE Xplorer, соответственно, а в качестве конкретных статей и результатов интеллектуальной деятельности отбирались наиболее релевантные запросу «машинное обучение, поиск уязвимостей». В каждом обзоре, по возможности, будут отражены следующие аспекты исследований: область применения (**Область**), метод поиска (**Метод**), решаемая МО задача (**Задача**), реализация МО (**Реализация**) и степень готовности решения (**Степень**).

Работа [6] посвящена безопасности Web-приложений (**Область**), в интересах чего авторы описывают разработанный сканер уязвимостей – ВТВ (*аббр. от англ. Bug Terminating Bot, перев. на русс. – бот для устранения ошибок*). Суть работы ВТВ заключается в сканировании страниц Web-приложений и тестировании их работоспособности под «полезной» нагрузкой (например, осуществляя SQL-инъекцию путем добавления специальных символов в конец запросов); возникновение нештатных ситуаций, таких, как ошибка выполнения, говорят и о наличии уязвимости (**Метод**). Также в ВТВ присутствует компонент исправления уязвимостей, предлагая необходимый для этого код. ВТВ имеет вид прототипа, разработанного на языке C# и использующего SQL-сервер в качестве собственной базы данных (БД). Применяемое авторами МО построено на базе SVM (**Реализация**) (*аббр. от англ. Support Vector Machine, перев. на русс. машины опорных векторов*) и используется на специальном сервере продукта (**Степень**) для повышения эффективности сканера, предлагая ему улучшенные параметры работы. Задачами МО, решаемыми в ВТВ, являются классификация и регрессионный анализ (**Задача**).

В аналитическом исследовании [7] оценивается текущее состояние поиска программных уязвимостей (**Область**) с применением МО, а также делаются прогнозы на будущее. В работе выделяются различные представления и векторизации кода программ (например, текстового ассемблерного,

полученного из бинарного машинного), состоящие из следующих: метрики программного инжиниринга, языковые модели описания кода, синтаксическое дерево (еще называемое деревом абстрактного синтаксиса, *перев. на англ.* Abstract Syntax Tree, *аббр.* AST) с семантической информацией, различные графы (потока управления, межпрограммных зависимостей и т. п.).

В качестве методов МО рассматриваются 2 типа-антагониста: контролируемые (в случае наличия пометок об уязвимых местах) и неконтролируемые (когда информация об уязвимостях изначально отсутствует) (**Метод**). В качестве реализации методов первого типа приводятся примеры применения Наивного Байеса, SVM, случайного леса, CNN (*аббр. от англ.* Convolutional Neural Networks, *перев. на русс.* сверточная нейронная сеть, модели последовательности, BLSTM (*аббр. от англ.* Bidirectional Long Short-Term Memory, *перев. на русс.* нейронные сети с двунаправленной долговременной краткосрочной памятью) и др.; указывается, что решения из двух последних примеров превосходят остальные (**Реализация**). В основу методов второго типа заложен поиск уязвимостей по шаблонам (**Метод**), созданным экспертным способом. Для этого в части МО применяется понижение размерности, кластеризация API-вызовов, выявление аномалий, а также анализ искажений (**Задача**) (например, оценка влияния переменных, полученных из испорченных пользовательских, на поведение системы).

В выводах статьи подчеркивается высокий потенциал методов поиска уязвимостей на базе МО; также приводятся новые возможные подходы (**Степень**), основанные на символьном выполнении, скрытых моделях Маркова и RNN (*аббр. от англ.* Recurrent Neural Network, рекуррентные нейронные сети).

Работа [8] посвящена разработке сканеров выявления SQL-инъекций (**Метод**) для Web-сайтов (**Область**). Для реализации МО в части классификации вредоносных SQL-команд (**Задача**) протестированы модели Наивного Байеса, SVM, дерево решений, случайный лес (**Реализация**), а также библиотеки XGBoost и CatBoost. Обучение производилось на наборах данных, содержащих валидные и небезопасные команды. Указывается, что предварительные эксперименты говорят о значении F-меры около 0.95, что, безусловно, является высоким показателем (**Степень**).

В статье [9] решается задача выявления уязвимостей типа SQL-инъекций, содержащихся в коде языка программирования PHP (**Область**). В интересах применения МО автор с помощью собственного скрипта (**Степень**) выявляет некоторые признаки исследуемого кода (**Метод**), содержащие наличие проверок и очисток ввода (актуальных

для противодействия данной уязвимости) – как стандартные, так и оригинальные. Для обработки признаков при подаче их формализованного представления на вход модели МО применялись две техники – 1) «мешок слов», позволяющий избавляться от терминов в коде, не связанных с SQL-инъекциями; 2) Word2Vec, учитывающая контекст функций [10]. Для определения наиболее эффективной реализации МО автор проводит эксперименты, применяя различные классификаторы (**Задача**): для 1-й техники – дерево решений, случайный лес, SVM, логистическая регрессия, MLP (*аббр. от англ.* Multi-Layer Perceptron, *перев. на русс.* многоуровневый перцептрон или перцептрон), RNN, BLSTM и CNN; для 2-й техники – последние 4 классификатора из первой техники (**Реализация**). Исходя из важности для настоящего исследования полученных результатов, они продублированы в таблице 1.

ТАБЛИЦА 1. Результаты экспериментов по классификации SQL-инъекций

TABLE 1. Experimental Results of SQL Injection Categorization

Классификатор	Полнота	Точность	F-мера
<i>для «мешка слов»</i>			
Дерево решений	56.5 %	93.4 %	0.650
Случайный лес	57.7 %	93.6 %	0.660
SVM	58.3 %	95.4 %	0.732
Логистическая регрессия	56.0 %	95.1 %	0.713
MLP	63.7 %	95.3 %	0.746
RNN	62.4 %	95.3 %	0.742
BLSTM	61.4 %	95.2 %	0.734
CNN	59.9 %	95.3 %	0.734
<i>для Word2Vec</i>			
MLP	50.7 %	92.8 %	0.601
RNN	49.9 %	92.7 %	0.595
BLSTM	48.5 %	93.2 %	0.606
CNN	48.1 %	92.5 %	0.573

Согласно полученным результатам (см. таблицу 1), делаются следующие выводы. Во-первых, F-мера определения уязвимостей оказалась выше при использовании «мешка слов». Во-вторых, наилучшего результата в случае «мешка слов» достиг MLP, а в случае Word2Vec – BLSTM. В-третьих, работа классификаторов глубокого обучения (т. е. последних четырех) для каждой техники оказалась лучше классических. И, в-четвертых, необходимо отметить низкое значение полноты (менее 66 %), что негативно влияет на количество выявленных уязвимостей.

В эмпирическом исследовании [11] указывается, что модели обнаружения уязвимостей в программном коде (**Область**) на основе глубокого обучения достигли значительного прогресса, пре-

вышающего в ряде случаев даже инструменты статического анализа. Тем не менее, понимание возможностей таких моделей еще далеко от конечной точки; в связи с чем в данной работе делаются экспериментальные оценки границ их применимости (**Степень**). В частности, авторы стремятся ответить на следующие вопросы:

- 1) Согласуются ли модели с реально обнаруженными уязвимостями?
- 2) Как на результативность обнаружения влияет тип уязвимости?
- 3) Сложнее ли делать предсказания для программ по определенным признакам кода?
- 4) Влияет ли увеличение датасета на эффективность поиска?
- 5) Как состав обучающих данных влияет на эту эффективность?
- 6) Какая информация об исходном коде использовалась моделями для предсказания уязвимостей?

Для ответа на вопросы авторы в своей работе воспроизвели 11 моделей глубокого обучения, построенных на абстрактных архитектурах (GNN, RNN, BLSTM, CNN и «Transformer» или преобразователь данных (**Реализация**) и обучаемых на конкретных датасетах. Модели применялись для классификации экземпляров кода на уязвимые и безопасные (**Задача**), используя для этого его различные признаки (по AST, графам потоков управления, зависимостям данных и др.) (**Метод**). Ниже все эти модели перечислены в нотации «название / архитектура / датасет(ы)»:

- Devign / GNN (*аббр. от англ. Graph Neural Network, перев. на русс. графовая нейронная сеть / Devign*;
- ReVeal / GNN, граф свойств / Devign, ReVeal;
- ReGVD / GNN, токен / Devign;
- CodeBERT / преобразователь данных / Devign;
- VulBERTa-CNN / преобразователь данных, CNN / VulDeePecker, Draper, ReVeal, μ VulDeePecker, Devign, D2A;
- VulBERTa-MLP / преобразователь данных / MLP;
- PLBART / преобразователь данных / Devign;
- LineVul / преобразователь данных / MSR (*аббр. от англ. Multi-Sentence Resampling*);
- Code2Vec / MLP, AST / Devign;
- SeSyVR / RNN / SARD (*аббр. от англ. Software Assurance Reference Dataset*), NVD (*аббр. от англ. National Vulnerability Database*);
- VulDeeLocator / BLSTM / SARD, NVD.

Были получены следующие ответы на соответствующие вопросы, поставленные исследователями:

- 1) 34,9 % тестовых данных (или 30,6 % общих данных) дают разные прогнозы в зависимости от используемых при обучении моделей.
- 2) Все модели с разной эффективностью способны предсказывать разные типы уязвимостей (в

экспериментах учитывались следующие из них: переполнение буфера, повышение привилегий, а также ошибки значений, ресурсов и проверки ввода); так, например, Devign и ReVeal предсказывали все типы уязвимостей с близкой вероятностью, для других же решений ошибки ресурсов обнаруживались хуже всего.

- 3) Предсказания всех моделей оказались лучше для простого набора данных (т. е. признаков кода), чем для сложного, с разницей в 10 %.

- 4) Увеличение датасета улучшает эффективность предсказания уязвимостей, но незначительно – при изменении набора данных от 10 % до 100 % от исходного, F-мера увеличилась на 0.16.

- 5) Разнообразие обучающих данных из разных проектов не дает преимуществ по сравнению с использованием данных только из одного проекта (для примера был взят набор Chrome из датасета MSR, содержащий 76 тысяч образцов).

- 6) Авторы получили матрицу схожести важных признаков для парного применения моделей (используя простое пересечение и меру Жаккара); так, хотя предсказания моделей имеют различную эффективность, однако используемая ими информация о коде пересекается.

Одной из основных предпосылок исследования [12] является автоматизация поиска уязвимостей в различных средах: сетях, Web-сайтах, программном обеспечении. В качестве обоснования к применению МО указаны сложность типового исследуемого объекта и невозможность создания полной базы всех существующих уязвимостей. Процесс поиска уязвимостей авторами описывается с помощью модели на базе марковского процесса принятия решения (**Метод**), когда под состоянием системы понимаются сведения об объекте исследования, а применение утилиты – как оптимизация действий по сбору дополнительных данных (**Задача**). В интересах этого предлагается применять соответствующую модель МО с подкреплением; в частности, на базе алгоритма Q-обучения (**Реализация**), характерного для агентного подхода. Описанные предложения были оформлены в виде программного прототипа (**Степень**) модели поиска уязвимостей Web-ресурсов (**Область**). Действиями по сбору информации являлись основные функции сканеров Web-сайтов – nmap, SQLmap, WhatWeb и XSScrapy. Состояние системы описывалось с помощью 128 элементов, включающих версию Web-сервера, его поддомены, XSS уязвимости и др. Используя датасет различных характеристик сайтов, модель была успешно обучена за 60 эпох – полных итераций по набору данных в процессе обучения. Также, авторы подчеркивают незаконченность предложенного решения, поскольку для реального его применения количество характеристик должно быть, по их мнению, примерно в 100 раз больше.

В другой аналитической работе [13], проведенной одним из соавторов настоящей статьи, осуществлена систематизация основных этапов статического анализа кода программ (**Область**) в интересах поиска уязвимостей с позиции применимости решения различных задач МО (**Метод**). Так, выделены следующие этапы поиска: сбор данных, подготовка объекта, проведение эксперимента и формирование результатов. С другой стороны, рассматриваются следующие задачи МО: классификация, выявление аномалий, регрессия, кластеризация и обобщение (**Задача**). Для более половины комбинаций этапов и задач приведены примеры обоснованности решений в интересах поиска. Так, например, классификация частей кода на легальные и небезопасные позволит обнаруживать существующие и близкие к ним уязвимости; а кластеризация, отнесение последних к отдельным классам. В результате была предложена обобщенная, с методологических позиций, модель (**Степень**), систематизирующая решения частных задач статического анализа в выбранной авторами статьи предметной области с применением МО. Какие-либо конкретные предложения по реализации решения задач МО в статье отсутствуют (**Реализация**).

В [14] поиск уязвимости в программах, имеющих форму машинного кода (**Область**), предлагается осуществлять с применением глубокой RNN для BLSTM (**Реализация**). При этом уязвимость интерпретируется, как некоторые абстрактные особенности кода – семантики инструкций, их влияния на регистры и др. Само МО применяется для решения типовой задачи классификации (**Задача**) исходных образцов программ на безопасные или содержащие уязвимости. Метод построения классификатора состоит из следующих шагов (**Метод**): 1) преобразование инструкций машинного кода в векторное представление, используя обучение с переносом [15], что позволяет делать семантически близкие инструкции, имеющими подобные представления; 2) построение RNN, входными признаками которой являются векторные представления инструкций; 3) фиксирование модели, готовой для применения. Авторы заявляют о выявлении уязвимости типа целочисленного переполнения с 84 %-ной точностью (**Степень**).

Направлением исследования в [16] является противодействие уязвимостям в локальных сетях и Web-приложениях (**Область**). Изначально указывается, что сфера информационных технологий стала одной из первых, в которые стали внедряться технологии искусственного интеллекта (ИИ); одной из причин этого авторы называют огромный объем данных, «неподъемный» для человека. Упоминаются некоторые системы автоматического поиска уязвимостей на базе ИИ: Nessus и OpenVAS – для локальных сетей, SQL-map и what-web –

для Web-ресурсов, антивирус Kaspersky – для ПО. Подчеркивается, что само по себе внедрение такого рода интеллектуальных систем может служить причиной появления новых уязвимостей из-за их собственной небезопасности; также, существуют и атаки на сами системы. Отмечаются успехи, достигнутые с помощью МО, построенного на глубоком обучении и сложных нейронных сетях. Сам же методологический подход (**Степень**) к автоматическому поиску уязвимостей описан с помощью следующих этапов: изучение процесса обнаружения уязвимостей, построение соответствующих моделей, их обучение и анализ результатов решения задачи поиска. МО авторы предлагают реализовывать на базе модели с подкреплением, и, в частности, с применением Q-обучения (**Реализация**), что позволит оптимизировать действия по поиску уязвимостей, аналогичные предпринимаемым человеком (**Задача**). Также приводится мнение экспертов касательно того, что ИИ, скорее всего, должен иметь рекомендательный характер, а окончательное решение касательно потенциальных уязвимостей должно приниматься экспертом (**Метод**).

Изобретение [17] содержит технический результат в виде повышения вероятности обнаружения уязвимостей в ПО (**Область**) за счет следующих действий (**Метод**): построения векторов атак на уязвимости; сбор датасета с «зараженным» кодом; построение AST и поиск в его элементах вектора атаки; составление путей между элементами; применение обученной на векторах атак модели МО для путей; итоговая классификация (**Задача**) уязвимостей с помощью алгоритма случайного леса, дерева решений или SVM (**Реализация**). Таким образом, хотя изобретение предполагает лишь результат творческой деятельности (**Степень**), однако логичность предложенных действий поиска уязвимостей позволяет говорить об их практической реализуемости.

Также в российском сегменте Интернета было найдено несколько свидетельств о регистрации программы для ЭВМ:

– в [18] содержится описание программы (**Степень**) для предсказания путем классификации (**Задача**) CVSS-метрик уязвимостей из базы данных (**Метод**) в формате CVE (**Область**); также оцениваются характеристики такого предсказания (точность, полнота и F-мера) на базе различных методов МО (**Реализация**) с помощью «исчерпывающего поиска кросс-валидации» [19];

Примечание: Кросс-валидация (перекрестная проверка) – это метод оценки моделей МО путем обучения нескольких из них на подмножествах доступных входных данных и их оценки на другом дополнительном подмножестве; используется для обнаружения неспособности распознать паттерн или шаблон.

– в [20] содержится решение по автоматическому поиску уязвимостей (**Степень**) в устройстве с сетевым доступом (**Область**), для чего о нем собирается информация, осуществляется поиск уязвимостей по БД (**Метод**), а также делается попытка их эксплуатации. Какие-либо детали касательно применения МО в реферате свидетельства отсутствуют (**Задача, Реализация**).

Поиск научных публикаций по указанному выше ключевому запросу показал критически малое количество отечественных исследований по сравнению с зарубежными; так, проанализированные в [7] релевантные работы российских ученых составляют большую часть из всех, найденных в РИНЦ.

Систематизация результатов

Осуществим систематизацию 12 проведенных обзоров путем их характеризования с помощью следующего набора критериев:

К_1 – год публикации, позволяющий оценить тенденции в предметной области;

К_2 – область применения решения, заявленно-го авторами;

К_3 – основная идея метода поиска уязвимостей, описанная в абстрактных терминах (для сравнения с другими);

К_4 – стандартная задача, решаемая с помощью МО и лежащая в основе метода поиска: классификация, регрессия, выявление аномалий, кластеризация, обобщение, оптимизация действий;

К_5 – предлагаемая реализация МО: конкретный алгоритм, модель и т. п. (без учета ансамблевых метаалгоритмов); в случае их большого количества будем указывать наиболее эффективные (при наличии такой информации);

К_6 – степень реализации изложенных в публикации результатов: теоретическое предположение (теор.), доведение теории до состояния работающего прототипа (прот.) или полностью готовое практическое решение (практ.).

Результаты систематизации представлены в таблице 2 (использованы следующие сокращения: послед. – последовательность; множ. – множество; «---» – информация по критерию отсутствует; в скобках может указываться некоторая особенность соответствия критериям).

ТАБЛИЦА 2. Результаты систематизации обзоров научных публикаций

TABLE 2. Results of Systematizing Reviewing Scientific Publications

Название, ссылка	К_1	К_2	К_3	К_4	К_5	К_6
Automatic Detection and Correction of Vulnerabilities using Machine Learning [6]	2017	Web-сайты, БД	Сканирование и тестирование образца	регрессия, кластеризация	SVM	практ.
Current and Future Research of Machine Learning Based Vulnerability Detection [7]	2018	Программное обеспечение	Выделение признаков и поиск по ним	классификация, выявление аномалий, обобщение	Модель послед., BLSTM	теор.
Determining Web Application Vulnerabilities Using Machine Learning Methods [8]	2023	Web-сайты, БД	Выделение признаков и поиск по ним	классификация	Наивный Байес, SVM, дерево решений, случайный лес	практ.
A Machine Learning Based Approach to Identify SQL Injection Vulnerabilities [9]	2019	Программное обеспечение, БД	Выделение признаков и поиск по ним	классификация	MLP, RNN, BLSTM, CNN	прот.
An Empirical Study of High-Impact Factors for Machine Learning-Based Vulnerability Detection [11]	2020	Программное обеспечение	Выделение признаков и поиск по ним	классификация	GNN, RNN, BLSTM, CNN, преобразователь данных	прот.
Применение машинного обучения с подкреплением для автоматизированного поиска уязвимостей информационных систем [12]	2020	Web-сайты	Использование марковского процесса принятия решения	оптимизация действий	Модель МО с подкреплением (Q-обучение)	прот.
Обобщенная модель статического анализа программного кода на базе машинного обучения применительно к задаче поиска уязвимостей [13]	2020	Программное обеспечение	«Отражение» этапов поиска на задачи МО	Все	---	теор.
Поиск уязвимостей в машинном коде с помощью методов глубокого обучения [14]	2018	Программное обеспечение	Выделение признаков и поиск по ним	классификация	BLSTM	прот.

Название, ссылка	K_1	K_2	K_3	K_4	K_5	K_6
Использование ИИ в поиске уязвимостей в локальных сетях или веб-приложениях [16]	2023	Web-сайты, локальные сети	Рекомендательный характер (без детализации решения)	оптимизация действий	Модель МО с подкреплением (Q-обучение)	теор.
Способ и система выявления эксплуатируемых уязвимостей в программном коде [17]	2023	Программное обеспечение	Выделение признаков и поиск по ним	классификация	Случайный лес, дерево решений, SVM	теор.
Компонент анализа эффективности методов машинного обучения для предсказания значений метрик уязвимостей [18]	2023	Базы уязвимостей	Предсказание по БД	классификация	--- (множ.)	прот.
Прототип программного решения, реализующий перспективные технологии искусственного интеллекта применительно к тестированию на проникновение информационных систем [20]	2022	Сетевые приложения	Предсказание по БД	---	---	прот.

Исходя из результатов сравнительного анализа сделанных обзоров (см. таблицу 2), можно сделать следующие выводы по каждому из критериев. При этом критерии K_2 и K_4 будут оценены по балльной системе следующим образом – каждый 1 балл будет делиться между всеми значениями по критерию для одной публикации (например, значение «Web-сайты» по K_2 для [6] будет иметь 0.5 балла, а для [12] – 1 балл).

Во-первых, публикационная активность имеет характерный возрастающий тренд (несмотря на отдельные «провалы» в 2019 и 2022 годах), притом начавшийся в последние 5-10 лет; это можно видеть на гистограмме распределения количества проанализированных релевантных публикаций за соответствующий год (рисунок 1) с линейной аппроксимацией.

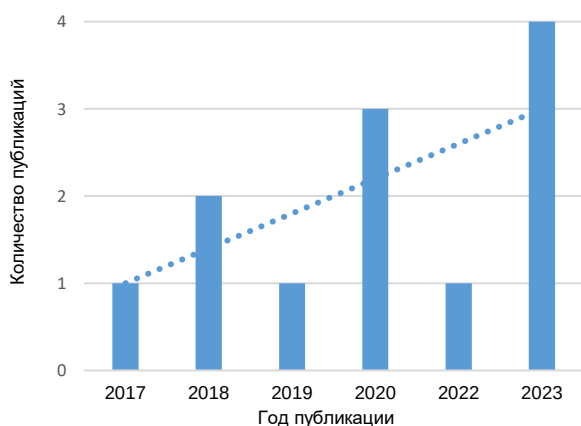


Рис. 1. Распределение количества публикаций по годам

Fig. 1. Distribution of the Number of Publications by Year

Можно предположить, что в будущем или тренд продолжит рост, поскольку необходимость выхода из «научно-технического тупика» по поиску уяз-

вимостей станет еще актуальней, или наоборот – резкий спад, если будет найдено универсальное решение на базе МО (что представляется менее вероятным).

Во-вторых, областями, для которых авторы исследований предлагают применять свои решения, являются следующие (в порядке убывания «популярности», соответствующей баллам и указанной после рейтинга R):

- Программное обеспечение – $R = 5.5$;
- Web-сайты – $R = 2.5$;
- БД – $R = 1.5$;
- Сетевые приложения – $R = 1.0$;
- Локальные сети, Базы уязвимостей – $R = 0.5$

Таким образом, международная (как отечественная, так и зарубежная) научная общественность сходит в мнении, что применимость МО для поиска уязвимостей в ПО наиболее востребована.

В-третьих, сравнение идей (с учетом сложности такого процесса) позволяет поставить на первое место среди остальных выделение некоторых признаков в объекте исследования (например, поддеревьев AST для исходного кода) и осуществление по ним поиска уязвимостей, в основном, с помощью классификаторов МО – в 6 публикациях. Остальные же используемые идеи носят частный характер и не могут быть обобщены – например, применение марковских процессов [12], метрик из БД [18, 20], подход «серого» или «черного ящика» [6].

Т. е., каких-либо нестандартных и широко применяемых подходов для поиска уязвимостей обнаружено не было, поскольку в основном используется классическое сопоставление признаков с некоторым шаблоном (точным или вероятностным).

В-четвертых, в 1 публикации не указаны решаемые задачи, характерные для МО, а для 1 – указаны все. По остальным же работам распределение популярности, согласно баллам и указанной после рейтинга R , следующее:

- классификация – $R = 6\frac{1}{3}$;
- оптимизация действий – $R = 2$;
- регрессия, кластеризация – $R = 0.5$;
- обобщение, выявление аномалий – $R = \frac{1}{3}$.

Таким образом, наиболее часто применяемым способом в части МО для поиска уязвимостей является классификация. Впрочем, исходя из частоты решения задачи оптимизации действий (с помощью МО с подкреплением на базе Q -алгоритма), данный способ также может быть рассмотрен, как перспективный. Решение же других задач может оказаться востребованным для вспомогательных целей.

В-пятых, какого-либо существенного преимущества реализации МО не было выявлено, а их баллы распределились следующим образом: Модель МО с подкреплением (Q -обучение) – 2, LSTM (включая BLSTM) – 1.95, SVM – 1.5 балла, остальные – менее 0.5. Таким образом, можно говорить о выделении Q -алгоритмов и (B)LSTM-нейронной сети, как достаточно перспективных реализаций решения задачи поиска уязвимостей в ПО (хотя, следуя обзорам [12] и [16], Q -алгоритм не связан с этим напрямую). Как видно, все подходы к реализации МО имеют право на существование.

И, в-шестых, в 4-х работах описаны лишь теоретические выкладки, в 6-ти – предложен реально действующий прототип, и только в 2-х – исследование доведено до полноценного практического решения. Скорее всего, это означает, что или при переходе от прототипа к продукту возникают принципиальные сложности реализации, или же достигнутая эффективность поиска уязвимостей оказывается недостаточной для реального применения.

Концептуальная система поиска уязвимостей

На основании сделанного обзора публикаций, их систематизации, а также авторского опыта, предлагается (новая) концептуальная система поиска уязвимостей в программном обеспечении с применением МО (далее – Система), используемая экспертом по безопасности ПО (далее – Эксперт) и основанная на следующих принципах.

Принцип 1. В Системе должно применяться многостороннее (или многофункциональное) исследование образца программы: статический и динамический анализ, тестирование потенциально опасных мест, прогнозирование уязвимостей по метрикам и т. п.

Принцип 2. Должно использоваться сложное (многоэтапное, иерархическое, итеративное и т. п.) комбинирование известных методов сбора информации и поиска по ней уязвимостей.

Принцип 3. В основе всех частей-компонентов Системы (т. е. ее методов и управления ими) должно лежать МО.

Принцип 4. В случае невозможности гарантированного определения уязвимостей, их признаков, связанных с ними частей программы, Система может корректировать свою работу с помощью Эксперта (таким образом, Система относится к разряду поддерживающих принятие решений человеком).

Принцип 5. Информация, накапливаемая и используемая Системой, должна находиться в локальной базе данных, синхронизируемой с внешними хранилищами.

Принцип 6. Система должна давать рекомендации Эксперту касательно потенциальных уязвимостей.

Принцип 7. Все действия в Системе должны происходить на единой программно-аппаратной платформе в тесной связи друг с другом.

Таким образом, основная идея Системы заключается во всестороннем исследовании объекта (т. е. программы) различными базовыми методами, управляемыми настраиваемым метаалгоритмом. Вся собираемая информация при невозможности ее автоматической обработки должна корректироваться Экспертом, а результаты работы касательно найденных уязвимостей должны иметь рекомендательный характер, позволяя тем самым корректировать настройки Системы для повторного запуска. Также, метрики и другая информация об уязвимости должны браться не только из локальной БД (накапливая в результате работы), но и актуализироваться путем синхронизации с внешними БД (например, банком данных угроз ФСТЭК, NVD, CVE и др.).

Схема предлагаемой концептуальной Системы представлена на рисунке 2, где использованы следующие цветовые и иные обозначения: зеленый – объект исследования, синий – методы исследования, серый (с оттенками) – группировка или контейнер для элементов, оранжевый (с оттенками) – получаемые данные, красный – результат работы (т. е. уязвимости); желтый – использование МО; пометка «...» означает возможность расширения элементов группы.

Хотя схема является интуитивно понятной, тем не менее, дадим несколько принципиальных пояснений к ней путем указания соответствия принципов и частей схемы.

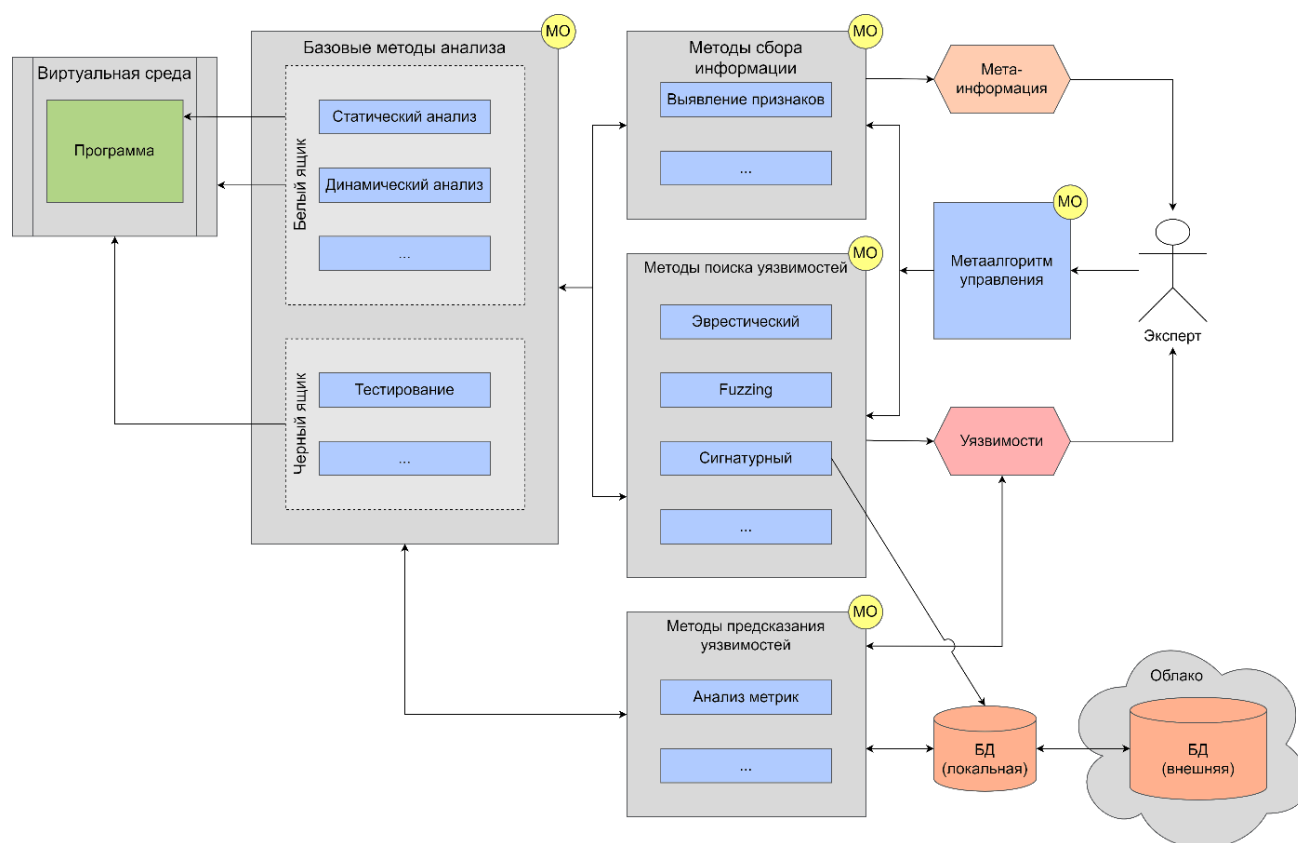


Рис. 2. Концептуальная система поиска уязвимостей в ПО с применением МО

Fig 2. A Conceptual System for Software Vulnerability Search Using Machine Learning

Принцип 1 учтен тем, что базовые методы анализа могут иметь различные походы («белый» и «черный ящик», а также их комбинацию), каждый из которых реализуем целым набором «инструментов» (статический и динамический анализ, тестирование выполнением и т. п.). При этом базовые методы могут взаимодействовать с объектом исследования напрямую (например, анализировать код программы) или через виртуальную среду (например, эмулируя выполнение программы).

Принцип 2 отражен с помощью комбинирования базовых методов при выполнении основных – сбора информации, поиска по ней уязвимостей, а также их предсказания (в том числе, по отдельным метрикам), которые могут оперировать целым набором базовых методов. Также, на схеме присутствует метаалгоритм, представляющий собой сложное решение по управлению другими методами, при этом в основе которого лежат не только эмпирические правила, но и обучаемые и тренируемые модели МО.

Выполнение Принципа 3 подтверждается тем, что практически все автоматические действия в Системе (кроме служебных, типа работы с БД) выполняются с применением МО (возможность чего была подтверждена в результатах рассмотренных исследований).

Соблюдение Принципа 4 достигается за счет прямого участия Эксперта в работе Системы путем предоставления ему собранной метаинформации о программе (в особенности, если такая не может дальше обрабатываться с необходимой точностью и полнотой), выявленных уязвимостях, их предсказаний на основании анализа обнаруженных или открытых метрик и т. п.

Для соответствия Принципу 5 в Системе предназначается локальная БД, синхронизируемая с глобальной (т. е. в «облаке»). Поскольку в основном в обеих БД хранится информация об уязвимостях (в том числе, их признаках в коде), то основное взаимодействие с БД осуществляют методы предсказания уязвимостей (например, по метрикам) и часть методов поиска уязвимостей (например, сигнатурный).

Поскольку согласно Принципу 6 Система предоставляет Эксперту лишь рекомендации (в смысле – вероятностные предположения) касательно результатов своей работы, то выходом автоматической части системы являются уязвимости, которые обрабатываются Экспертом и позволяют ему внести корректировки в алгоритмы работы Системы.

И наконец, все объекты, промежуточные данные, методы, БД (кроме глобальной) и сам Эксперт представляют собой взаимоувязанный человеко-управляемый программно-аппаратный комплекс по решению сложной научно-технической задачи, что подтверждает соответствие Принципу 7.

Заключение

В работе произведен обзор научных публикаций касательно возможностей применения МО для поиска уязвимостей в программах. Произведен сравнительный анализ обзоров (и их результатов), который позволил сделать систематизированные выводы касательно:

- актуализации исследования таких возможностей в последние годы;
- применимости к различным областям (хотя основной и является программное обеспечение);
- осуществления поиска уязвимостей (в основном, по их признакам в коде);
- решения задачи классификации вкупе с остальными, классическими для МО;
- преимуществ от применения той или иной реализации моделей и методов МО (установлено, что они отсутствуют);

Список источников

1. Романов Н.Е., Израилов К.Е., Покусов В.В. Система поддержки интеллектуального программирования: машинное обучение feat. быстрая разработка безопасных программ // Информатизация и связь. 2021. № 5. С. 7–17. DOI:10.34219/2078-8320-2021-12-5-7-16
2. Chavan A., Pimplikar S., Deshmukh A. An Overview of Machine Learning Techniques for Evaluation of Pavement Condition // Proceedings of the 4th International Conference on Cybernetics, Cognition and Machine Learning Applications (ICCCMLA, Goa, India, 08–09 October 2022). IEEE, 2022. PP. 139–143. DOI:10.1109/ICCCMLA56841.2022.9989164
3. Sathuluri M.R., Sahithi R., Sri P.N., Arshitha K. Machine Learning Approach to Design Fractal Antenna for 5G Applications // Proceedings of the 4th International Conference on Inventive Research in Computing Applications (ICIRCA, Coimbatore, India, 21–23 September 2022). IEEE, 2022. PP. 275–280. DOI:10.1109/ICIRCA54612.2022.9985480
4. Rana P., Gupta L. R., Dubey M.K., Kumar G. Review on evaluation techniques for better student learning outcomes using machine learning // Proceedings of the 2nd International Conference on Intelligent Engineering and Management (ICIEM, London, United Kingdom, 28–30 April 2021). IEEE, 2021. PP. 86–90. DOI:10.1109/ICIEM51511.2021.9445294
5. AlShehri Y., Ramaswamy L. SECOE: Alleviating Sensors Failure in Machine Learning-Coupled IoT Systems // Proceedings of the 21st International Conference on Machine Learning and Applications (ICMLA, Nassau, Bahamas, 2–14 December 2022). IEEE, 2022. PP. 743–747. DOI:10.1109/ICMLA55696.2022.00124
6. Tommy R., Sundeeep G., Jose H. Automatic Detection and Correction of Vulnerabilities using Machine Learning // Proceedings of the International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC, Mysore, India, 08–09 September 2017). IEEE, 2017. PP. 1062–1065. DOI:10.1109/CTCEEC.2017.8454995
7. Jin Z., Yu Y. Current and Future Research of Machine Learning Based Vulnerability Detection // Proceedings of the Eighth International Conference on Instrumentation & Measurement, Computer, Communication and Control (IMCCC, Harbin, China, 19–21 July 2018). IEEE, 2018. PP. 1562–1566. DOI:10.1109/IMCCC.2018.00322
8. Zhumabekova A., Matson E.T., Karyukin V., Zhumabekova K., Zhuandykov B., Ussatova O., et al. Determining Web Application Vulnerabilities Using Machine Learning Methods // Proceedings of the 19th International Asian School-Seminar on Optimization Problems of Complex Systems (OPCS, Novosibirsk, Moscow, Russian Federation, 14–22 August 2023). IEEE, 2023. PP. 136–139. DOI:10.1109/OPCS59592.2023.10275756
9. Zhang K. A Machine Learning Based Approach to Identify SQL Injection Vulnerabilities // Proceedings of the 34th IEEE/ACM International Conference on Automated Software Engineering (San Diego, USA, 11–15 November 2019). IEEE, 2019. PP. 1286–1288. DOI:10.1109/ASE.2019.00164
10. Le Q.V., Mikolov T. Distributed Representations of Sentences and Documents // Proceedings of the 31st International Conference on Machine Learning (PMLR, Beijing, China, 21–26 June 2014). 2014. Vol. 32. Iss. 2. PP. 1188–1196.
11. Zheng W., Gao J., Wu X., Xun Y., Liu G., Chen X. An Empirical Study of High-Impact Factors for Machine Learning-Based Vulnerability Detection // Proceedings of the IEEE 2nd International Workshop on Intelligent Bug Fixing (IBF, London, ON, Canada, 18–18 February 2020). IEEE, 2020. PP. 26–34. DOI:10.1109/IBF50092.2020.9034888

– степени доведения идей исследователей до практических решений.

Как известно авторам, подобный обзор релевантных научных публикаций в РИНЦ (совместно с международными базами), содержащий предметно-ориентированное критериальное сравнение, выполнен впервые.

Учет достоинств, недостатков и особенностей изложенных в них решений, позволил авторам аргументированно декларировать 7 принципов построения концептуальной системы поиска уязвимостей в программном обеспечении с применением МО, а также собственно и синтезировать ее схему. Ее теоретическая значимость заключается в систематизации научного знания предметной области, а практическая состоит в возможности проектирования архитектуры такой системы и предпосылках к непосредственной реализации соответствующего программного решения.

Продолжением исследования должна стать оценка возможности реализации отдельных элементов Системы имеющимся инструментарием (а в случае его отсутствия – разработка нового).

12. Выборнова О.Н., Рыжиков А.Н. Применение машинного обучения с подкреплением для автоматизированного поиска уязвимостей информационных систем // Математические методы в технике и технологиях – ММТТ. 2020. Т. 4. С. 110–113.
13. Буйневич М.В., Израйлов К.Е. Обобщенная модель статического анализа программного кода на базе машинного обучения применительно к задаче поиска уязвимостей // Информатизация и связь. 2020. №. 2. С. 143–152. DOI:10.34219/2078-8320-2020-11-2-143-152
14. Демидов Р.А. Поиск уязвимостей в машинном коде с помощью методов глубокого обучения // IV межрегиональная научно-практическая конференция «Перспективные направления развития отечественных информационных технологий» (Севастополь, Российская Федерация, 18–22 сентября 2018). Севастопольский государственный университет, 2018. С. 237–238.
15. Bottou L. From machine learning to machine reasoning // Machine Learning. 2014. Vol. 94. Iss. 2. PP. 133–149. DOI:10.1007/s10994-013-5335-x
16. Осман С.Ш.О. Использование ИИ в поиске уязвимостей в локальных сетях или Веб-приложениях // IX международная научно-практическая конференция «Актуальные аспекты развития науки и общества в эпоху цифровой трансформации (шифр-МКАА)» (Москва, Российская Федерация, 25 июля 2023). Махачкала: Издательство АЛЕФ, 2023. С. 83–88.
17. Максимова А.А., Гончаренко Л.Х., Бачевский А.Е., Гуртова К.С., Умеренко Г.С., Анистратенко М.А. Способ и Система выявления эксплуатируемых уязвимостей в программном коде. Патент на изобретение RUS 2790005 C1 от 10.03.2022. Оpubл. 14.02.2023.
18. Левшун Д.С. Компонент анализа эффективности методов машинного обучения для предсказания значений метрик уязвимостей. Свидетельство о регистрации программы для ЭВМ № RU 2023619249 от 05.05.2023. Оpubл. 05.05.2023.
19. Celisse A. Optimal cross-validation in density estimation with the L^2 -loss // The Annals of Statistics. 2014. Vol. 42. Iss. 5. PP. 1879–1910. DOI:10.1214/14-AOS1240
20. Кустаров Д.А., Сорокин Л.А., Трухачев А.А. Прототип программного решения, реализующий перспективные технологии искусственного интеллекта применительно к тестированию на проникновение информационных систем. Свидетельство о регистрации программы для ЭВМ № RU 2022682324 от 22.11.2022. Оpubл. 29.11.2022.

References

1. Romanov N.E., Izrailov K.E., Pokussov V.V. Intelligent Programming Support System: Machine Learning Feat. Fast Development of Secure Programs. *Informatization and communication*. 2021;5:7–17. DOI:10.34219/2078-8320-2021-12-5-7-16
2. Chavan A., Pimplikar S., Deshmukh A. An Overview of Machine Learning Techniques for Evaluation of Pavement Condition. *Proceedings of the 4th International Conference on Cybernetics, Cognition and Machine Learning Applications, ICCMLA, 08–09 October 2022, Goa, India*. IEEE; 2022. p.139–143. DOI:10.1109/ICCMLA56841.2022.9989164
3. Sathuluri M.R., Sahithi R., Sri P.N., Arshitha K. Machine Learning Approach to Design Fractal Antenna for 5G Applications. *Proceedings of the 4th International Conference on Inventive Research in Computing Applications, ICIRCA, 21–23 September 2022, Coimbatore, India*. IEEE; 2022. p.275–280. DOI:10.1109/ICIRCA54612.2022.9985480
4. Rana P., Gupta L. R., Dubey M.K., Kumar G. Review on evaluation techniques for better student learning outcomes using machine learning. *Proceedings of the 2nd International Conference on Intelligent Engineering and Management, ICIE, 28–30 April 2021, London, United Kingdom*. IEEE; 2021. p.86–90. DOI:10.1109/ICIE51511.2021.9445294
5. AlShehri Y., Ramaswamy L. SECOE: Alleviating Sensors Failure in Machine Learning-Coupled IoT Systems. *Proceedings of the 21st International Conference on Machine Learning and Applications, ICMLA, 2–14 December 2022, Nassau, Bahamas*. IEEE; 2022. p.743–747. DOI:10.1109/ICMLA55696.2022.00124
6. Tommy R., Sundeep G., Jose H. Automatic Detection and Correction of Vulnerabilities using Machine Learning. *Proceedings of the International Conference on Current Trends in Computer, Electrical, Electronics and Communication, CTCEEC, 08–09 September 2017, Mysore, India*. IEEE; 2017. p.1062–1065. DOI:10.1109/CTCEEC.2017.8454995
7. Jin Z., Yu Y. Current and Future Research of Machine Learning Based Vulnerability Detection. *Proceedings of the Eighth International Conference on Instrumentation & Measurement, Computer, Communication and Control, IMCCC, 19–21 July 2018, Harbin, China*. IEEE; 2018. p.1562–1566. DOI:10.1109/IMCCC.2018.00322
8. Zhumabekova A., Matson E.T., Karyukin V., Zhumabekova K., Zhuandykov B., Ussatova O., et al. Determining Web Application Vulnerabilities Using Machine Learning Methods. *Proceedings of the 19th International Asian School-Seminar on Optimization Problems of Complex Systems, OPCS, 14–22 August 2023, Novosibirsk, Moscow, Russian Federation*. IEEE; 2023. p.136–139. DOI:10.1109/OPCS59592.2023.10275756
9. Zhang K. A Machine Learning Based Approach to Identify SQL Injection Vulnerabilities. *Proceedings of the 34th IEEE/ACM International Conference on Automated Software Engineering, 11–15 November 2019, San Diego, USA*. IEEE; 2019. p.1286–1288. DOI:10.1109/ASE.2019.00164
10. Le Q.V., Mikolov T. Distributed Representations of Sentences and Documents. *Proceedings of the 31st International Conference on Machine Learning, PMLR, 21–26 June 2014, Beijing, China*. 2014;32(2):1188–1196.
11. Zheng W., Gao J., Wu X., Xun Y., Liu G., Chen X. An Empirical Study of High-Impact Factors for Machine Learning-Based Vulnerability Detection. *Proceedings of the IEEE 2nd International Workshop on Intelligent Bug Fixing, IBF, 18–18 February 2020, London, ON, Canada*. IEEE; 2020. p.26–34. DOI:10.1109/IBF50092.2020.9034888
12. Vybornova O.N., Ryzhikov A.N. Reinforcement Learning for Automated Vulnerability Search. *Matematicheskie metody v tekhnike i tekhnologiiakh – MMTT*. 2020;4:110–113.

13. Buinevich M.V., Izrailov K.E. Generalized model of software code's static analysis based on machine learning for vulnerability search. *Informatization and communication*. 2020;2:143–152. DOI:10.34219/2078-8320-2020-11-2-143-152
14. Demidov R.A. Vulnerability Search in Machine Code Using Deep Learning Approach. Proceedings of the IVth interregional Scientific and Practical Conference on Promising Directions for the Development of Domestic Information Technologies, 18–22 September 2018, Sevastopol, Russian Federation. Sevastopol State University Publ.; 2018. p.237–238.
15. Bottou L. From machine learning to machine reasoning. *Machine Learning*. 2014;94(2):133–149. DOI:10.1007/s10994-013-5335-x
16. Osman S.Sh.O. Using AI in Searching for Vulnerabilities in Local Networks or Web Applications. *IX International Scientific and Practical Conference on Current Aspects of the Development of Science and Society in the Era of Digital Transformation, Code – MCAA, 25 July 2023, Moscow, Russian Federation*. Makhachkala: ALEF Publ., 2023. p.83–88.
17. Maksimova A.A., Goncharenko L.Kh., Bachevsky A.E., Gurtova K.S., Umerenko G.S., Anistratenko M.A. *Method and System for Identifying Exploitable Vulnerabilities in Program Code*. Patent RF, no. 2790005 C1, 14.02.2023.
18. Levshun D.S. *Component for Analyzing the Effectiveness of Machine Learning Methods for Predicting the Values of Vulnerability Metrics*. Patent RF, no. 2023619249, 05.05.2023.
19. Celisse A. Optimal cross-validation in density estimation with the L^2 -loss. *The Annals of Statistics*. 2014;42(5):1879–1910. DOI:10.1214/14-AOS1240
20. Kustarov D.A., Sorokin L.A., Trukhachev A.A. *A Prototype of a Software Solution That Implements Promising Artificial Intelligence Technologies in Relation to Penetration Testing Of Information Systems*. Patent RF, no. 2022682324, 29.11.2022.

Статья поступила в редакцию 01.12.2023; одобрена после рецензирования 14.12.2023; принята к публикации 18.12.2023.

The article was submitted 01.12.2023; approved after reviewing 14.12.2023; accepted for publication 18.12.2023.

Информация об авторе:

ЛЕОНОВ
Николай Викторович

кандидат технических наук, доцент, начальник лаборатории Государственного научно-исследовательского института прикладных проблем
 <https://orcid.org/0009-0005-1295-5343>

БУЙНЕВИЧ
Михаил Викторович

доктор технических наук, профессор, профессор кафедры прикладной математики и информационных технологий Санкт-Петербургского университета государственной противопожарной службы МЧС России
 <https://orcid.org/0000-0001-8146-0022>

Научная статья

УДК 621.39; 004.056

DOI:10.31854/1813-324X-2023-9-6-95-100



Характеристики уязвимости аппаратуры потребителей глобальных навигационных спутниковых систем к спуфинг-атакам

Валерий Владимирович Неровный¹, valery.km@yandex.ru

Павел Дмитриевич Коратаев¹, korataev2015@mail.ru

Пётр Сергеевич Облов¹, oblov1997@yandex.ru

Марина Юрьевна Толстых^{2,3}, marina_lion@mail.ru

¹Военно-воздушная академия им. проф. Н.Е. Жуковского и Ю.А. Гагарина», Воронеж, 394064, Российская Федерация

²Московский государственный лингвистический университет, Москва, 119034, Российская Федерация

³Московский университет МВД России им. В.Я. Кикотя, Москва, 117437, Российская Федерация

Аннотация: В статье рассмотрены подходы к оценке уязвимости навигационной аппаратуры потребителей глобальных навигационных спутниковых систем (НАП ГНСС) к спуфинг-атакам, а также технологии защиты от помех и спуфинга. Предложены подходы и средства для оценки уязвимости НАП ГНСС при наличии имитирующих помех. Полученные результаты могут быть использованы разработчиками при создании НАП ГНСС нового поколения с улучшенной защитой от спуфинга.

Ключевые слова: навигационная аппаратура потребителей глобальных навигационных спутниковых систем, спуфинг, помеха, угроза, моделирование, уязвимость

Ссылка для цитирования: Неровный В.В., Коратаев П.Д., Облов П.С., Толстых М.Ю. Характеристики уязвимости аппаратуры потребителей глобальных навигационных спутниковых систем к спуфинг-атакам // Труды учебных заведений связи. 2023. Т. 9. № 6. С. 95–100. DOI:10.31854/1813-324X-2023-9-6-95-100

Vulnerability Characteristics of Global Navigation Satellite System Consumer Equipment to Spoofing Attacks

Valery Nerovny¹, valery.km@yandex.ru

Pavel Korataev¹, korataev2015@mail.ru

Petr Oblov¹, oblov1997@yandex.ru

Marina Tolstykh^{2,3}, marina_lion@mail.ru

¹Military Educational and Scientific Center of the Air Force N.E. Zhukovsky and Y.A. Gagarin Air Force Academy, Voronezh, 394064, Russian Federation

²Moscow State Linguistic University, Moscow, 119034, Russian Federation

³Kikot Moscow University of the Ministry of Internal Affairs of Russia, Moscow, 117437, Russian Federation

Abstract: The article presents approaches to assessing the vulnerability of navigation equipment of global navigation satellite systems consumers to spoofing attacks, as well as technologies for protection against interference and spoofing. Approaches and tools are proposed for assessing the vulnerability of navigation equipment of global navigation satellite systems in the presence of simulated interference. The results obtained can be used by developers to create a new generation global navigation satellite systems with improved protection against spoofing.

Keywords: navigation equipment of consumers of global navigation satellite systems, spoofing, interference, threat, modeling, vulnerability

For citation: Nerovny V., Korataev P., Oblov P., Tolstykh M. Vulnerability Characteristics of Global Navigation Satellite System Consumer Equipment to Spoofing Attacks. *Proceedings of Telecommun. Univ.* 2023;9(6):95–100. DOI:10.31854/1813-324X-2023-9-6-95-100

Введение

Большинство современных систем перевозок и перемещений реализуются посредством поддержки глобальных навигационных спутниковых систем (ГНСС) для обеспечения точной навигации. Высокоточные приемники ГНСС на борту современных судов и транспортных средств (как в гражданской, так и военизированной областях) в большей степени заслуживают доверия. Однако ввиду динамичных в своем открытии и внедрении технологических разработок, а также вредоносной деятельности со стороны противников (злоумышленников) концепция априорного доверия к системам навигации, очевидно, становится не актуальной.

Известны множественные случаи непреднамеренной передачи ГНСС-сигналов, а также вмешательства в их прием (внутриполосные или внеполосные помехи) [1, 2]. Кроме того, набирает популярность практика обмана приемников ГНСС, сообщаящих ложные данные о местоположении или времени (спуфинг, от англ. Spoofing). При этом демонстрировалось, что подобные деструктивные манипуляции осуществлялись с помощью относительно недорогого оборудования и с наличием небольших экспертных знаний у атакующих. Фактически структура данных, схемы модуляции и коды расширения общедоступны, в совокупности данные особенности позволяют создавать помехи и подделку.

Отмеченное подтверждает факт снижения существующих защитных барьеров для злонамеренных атак с далеко идущими последствиями. Следовательно, априорное доверие к ГНСС не релевантно, необходимы методы надежного обнаружения, оценки и борьбы с помехами и спуфингом, осуществляемых в отношении ГНСС.

К настоящему моменту в мировых масштабах однозначно не определена унифицированная практика мониторинга спуфинга навигационных сигналов, также отсутствуют юридически закрепленные критерии оценки и анализа указанных инцидентов безопасности. Вместе с тем известен ряд работ, посвященных тематике спуфинга навигационной аппаратуры потребителей ГНСС (НАП ГНСС) [1–8]. В

связи с растущей зависимостью различных отраслей промышленности от ГНСС остро стоит вопрос обработки последствий перебоев в работе ГНСС-услуг, которые могут быть критическими с точки зрения обеспечения безопасности, превенции или локализации экономического, социального или экологического ущерба.

Таким образом, востребована разработка подходов к способам и средствам снижения уязвимости ГНСС к спуфинг-атакам, включая резервное копирование метаданных о навигации, позиционировании, времени, полное использование текущих программ модернизации ГНСС, мониторинг и расширение целостности систем, а также технологии защиты от помех и противодействия спуфингу.

Подходы к оценке уязвимости навигационной аппаратуры потребителей глобальных навигационных спутниковых систем к спуфинг-атакам

В зарубежной технической литературе термин Spoofing используется для обозначения имитирующих помех. Работы, посвященные его анализу и изучению в отношении НАП ГНСС, ведутся, в основном, по двум направлениям:

- формирование алгоритмов и технологий создания имитирующих помех НАП ГНСС [1, 2];
- разработка способов и мер защиты от имитирующих помех [5–7].

Однако остается открытым вопрос оценки уязвимости НАП ГНСС к спуфинг-атакам. Цель данной работы – определение характеристик уязвимости НАП ГНСС к спуфинг-атакам, которые смогут впоследствии дополнить и сформировать унифицированный методологический аппарат к обеспечению безопасности ГНСС, например, в системе менеджмента (управления) уязвимостями.

В современной НАП ГНСС оценка параметров навигационных сигналов в каналах обработки осуществляется в несколько стадий. На стартовой итерации происходит предварительная (грубая) оценка параметров навигационных сигналов, на втором этапе формируются точные текущие оценки параметров (задержки и доплеровского смещения частоты, ДСЧ) навигационных сигналов [3, 4]. Первый

этап реализован поиском навигационных сигналов по соответствующим параметрам, второй – использованием следящих систем за фазой, частотой и задержкой навигационных сигналов.

Необходимо отметить, что поиск навигационных сигналов определяется как задача оценки его задержки и ДСЧ, которые принимаются постоянными на интервале наблюдения. При поиске осуществляется перебор с заданным шагом всего диапазона значений задержки и ДСЧ. Для каждого значения задержки и ДСЧ в каналах обработки вычисляется огибающая, которая сравнивается с порогом и принимается решение о наличии сигнала. За оценку значений параметров принимаются значения задержки и ДСЧ, при которых произошло превышение порога. При отсутствии сигнала анализируются следующие значения параметров, а при его наличии происходит переключение канала обработки в режим слежения за задержкой и ДСЧ.

Аппаратная реализация поиска сигналов в аппаратуре потребителей глобальных навигационных спутниковых систем

При аппаратной реализации алгоритмов поиска сигналов в НАП ГНСС используются режимы «полного поиска» и «допоиска», что обусловлено количеством априорной информации о значениях задержки и ДСЧ принимаемого сигнала. Режим «полного поиска» применяется для случая, когда исходная информация о задержке и ДСЧ-сигнала представлена в недостаточном объеме, отсутствует или является достаточно грубой. Данная ситуация характерна для начального включения НАП ГНСС, когда не имеется никаких сведений о текущих эфемеридах и текущем времени в приемнике. Позиция «допоиска» задействована для случая, когда имеется достаточно априорных сведений о характеристиках задержки и ДСЧ с некоторой точностью, что возможно для следующих событий:

- при включении НАП ГНСС присутствуют исходные данные о показателях задержки и ДСЧ либо от внешних источников, либо информация генерируется из данных сохраненного с прошлого сеанса альманаха системы;
- после срыва слежения за сигналом с целью перезахвата сигнала и повторного вхождения в режим слежения;
- при приеме сигнала навигационного спутника из состава орбитальной группировки, появившегося из-за линии радиогоризонта в зоне видимости НАП ГНСС.

Для оценки уязвимости к спуфинг-атакам НАП ГНСС можно представить в виде обобщенной схемы (рисунок 1).

В работе [5] была проведена экспериментальная оценка качества функционирования НАП ГНСС в

условиях спуфинг-атаки при слежении за параметрами навигационных сигналов, а публикация [6] посвящена теоретической оценке качества функционирования НАП ГНСС в условиях спуфинг-атаки при поиске навигационных сигналов.

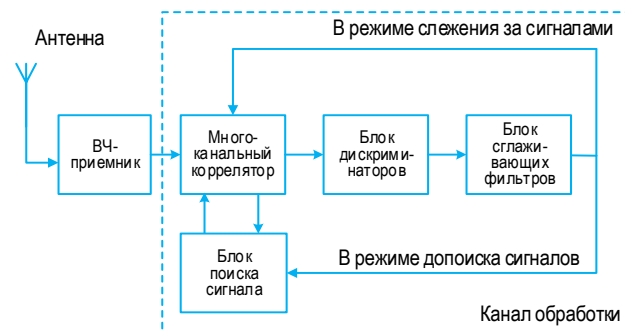


Рис. 1. Обобщенная схема НАП ГНСС для оценки уязвимости к спуфинг-атакам

Fig. 1. Generalized Diagram of Navigation Consumers Equipment of Global Navigation Satellite System (GNSS) for Assessing Vulnerability to Spoofing Attacks

Анализ, приведенных в работах [5, 6] результатов показывает, что захват следящими системами имитирующей помехи происходит при превышении помехи над сигналом не менее 20 дБ, а захват при поиске навигационных сигналов происходит при превышении помехи над сигналом не менее 2–4 дБ. Таким образом, можно сделать вывод о том, что наибольшая уязвимость НАП ГНСС к спуфинг-атаке будет при поиске навигационных сигналов. При этом одинаковой уязвимостью к спуфинг-атаке в НАП ГНСС будут обладать режимы «полного поиска» и «допоиска».

Ограничение задачи исследования и эксперимента

При выборе показателя уязвимости к спуфинг-атаке НАП ГНСС необходимо учитывать следующие ограничения и допущения:

- НАП ГНСС находится в зоне действия источника ложных навигационных сигналов;
- уровень имитирующих помех превышает уровень навигационных сигналов на 2–4 дБ;
- для успешного проведения спуфинг-атаки достаточно приема и обработки хотя бы одной помехи, имитирующей навигационные сигналы;
- в начальный при воздействии спуфинг-атаки момент НАП ГНСС осуществляет слежение за параметрами навигационных сигналов, а переключение в режим «допоиска» происходит при приеме сигнала одного или нескольких навигационных спутников из состава орбитальной группировки, появившегося из-за линии радиогоризонта в зоне видимости.

С учетом ограничений и допущений в качестве показателя целесообразно выбрать:

– время появления T_{nc} одного или нескольких навигационных спутников из состава орбитальной группировки из-за линии радиогоризонта в зоне видимости НАП ГНСС;

– вероятность уязвимости к спуфинг-атаке P_{ca} , которая определяется как вероятность появления за заданное время в зоне видимости НАП ГНСС одного или нескольких навигационных спутников.

Для наблюдателя, находящегося в области пространства, где размещена НАП ГНСС, время T_{nc} будет случайной величиной. Ради количественной оценки статистических характеристик величины T_{nc} (математического ожидания, дисперсии и функции распределения) была разработана программа для ЭВМ, в которой моделируется динамика изменения пространственной конфигурации созвездий различных ГНСС.

Работа программы моделирования изменений пространственной конфигурации созвездий ГНСС

Основу программы составляет модель невозмущенного движения спутников, которая задается математическими соотношениями [7]:

$$X = (R_3 + H_{nc}) \times (\cos \omega_n \cdot \cos \lambda - \sin \omega_n \cdot \sin \lambda \cdot \cos i), \quad (1)$$

$$Y = (R_3 + H_{nc}) \times (\cos \omega_n \cdot \cos \lambda + \sin \omega_n \cdot \sin \lambda \cdot \cos i), \quad (2)$$

$$Z = (R_3 + H_{nc}) \cdot \sin \omega_n \cdot \sin i, \quad (3)$$

$$\omega_n = \omega_0 + t_{тек} \cdot \frac{2\pi}{T_{cp}}, \quad (4)$$

$$\lambda = \lambda_0 + t_{тек} \cdot \frac{2\pi}{T_{cp}}, \quad (5)$$

где X, Y, Z – геоцентрические координаты навигационного спутника; H_{nc} – высота орбиты навигационного спутника; ω_0 – начальное значение Аргумента перигея; λ_0 – начальное значение долготы восходящего узла; R_3 – радиус Земли; i – текущее значение наклона орбиты навигационного спутника; T_{cp} – среднее значение драконического периода обращения навигационного спутника; ω_n – Аргумент перигея (рад); $t_{тек}$ – текущее время, на которое рассчитываются параметры движения спутника (сек.)

Исходными данными для программы являются геодезические координаты НАП ГНСС и вектора начальных геоцентрических координат навигационного спутника. В каждый момент времени в программе рассчитывается количество видимых спутников и время появления нового спутника в зоне видимости НАП ГНСС.

Для вновь появившегося i -го спутника определяется значение времени T_{nci} . После завершения работы программы формируется выборка значения времени T_{nc} объемом N для каждой ГНСС. После

ранжирования для каждой выборки определяются: выборочные средние, выборочные значения среднеквадратического отклонения и эмпирические функции распределения [8]:

$$\bar{T}_{nc} = \frac{1}{N} \sum_{i=1}^N T_{nci}, \quad (6)$$

$$\sigma_{T_{nc}} = \sqrt{\frac{1}{N} \sum_{i=1}^N (T_{nci} - \bar{T}_{nc})^2}, \quad (7)$$

$$F_n(T_{nc}) = \frac{n_{T_{nc}}}{N}, \quad (8)$$

где $\bar{T}_{nc}$ – выборочное среднее; $\sigma_{T_{nc}}$ – выборочное значение среднеквадратического отклонения; $F_n(T_{nc})$ – эмпирическая функция распределения; $n_{T_{nc}}$ – количество значений меньше T_{nc} .

Результаты моделирования

С использованием разработанной программы в среде MatLab/Simulink получены статистические характеристики случайной величины T_{nc} для ГНСС ГЛОНАСС, GPS, ГАЛИЛЕО и КОМПАСС, которые приведены в таблице 1, где N – объем выборки.

ТАБЛИЦА 1. Статистические характеристики T_{nc} для ГНСС ГЛОНАСС, GPS, ГАЛИЛЕО и КОМПАСС

TABLE 1. Statistical Characteristics of T_{nc} for GNSS GLONASS, GPS, GALILEO and COMPASS

Тип ГНСС	N	$\bar{T}_{nc}$, сек	$\sigma_{T_{nc}}$, сек
ГЛОНАСС	3408	2534	1497
GPS	4883	2065	1217
ГАЛИЛЕО	4251	2030	1447
КОМПАСС	4505	1916	1487

Эмпирические функции распределения случайной величины T_{nc} для ГНСС ГЛОНАСС, GPS, ГАЛИЛЕО и КОМПАСС приведены на рисунке 2.

Полученные эмпирические функции распределения позволяют определить вероятность уязвимости к спуфинг-атаке за заданное время как:

$$P_{ca}(T_{nc3} < T_{nc}) = F(T_{nc3}). \quad (9)$$

Для времени $T_{nc3} = 600$ сек вероятность уязвимости к спуфинг-атаке составила для НАП ГНСС: ГЛОНАСС – 0.075, GPS – 0.25, ГАЛИЛЕО – 0.18, КОМПАСС – 0.2.

Разброс значений вероятности уязвимости к спуфинг-атаке P_{ca} для НАП, осуществляющим функционирование по сигналам разных ГНСС, можно объяснить различным количеством навигационных спутников, а также пространственными конфигурациями орбитальных группировок.

Заключение

Сигналы ГНСС уязвимы к помехам, которые блокируют прием навигационных сообщений прием-

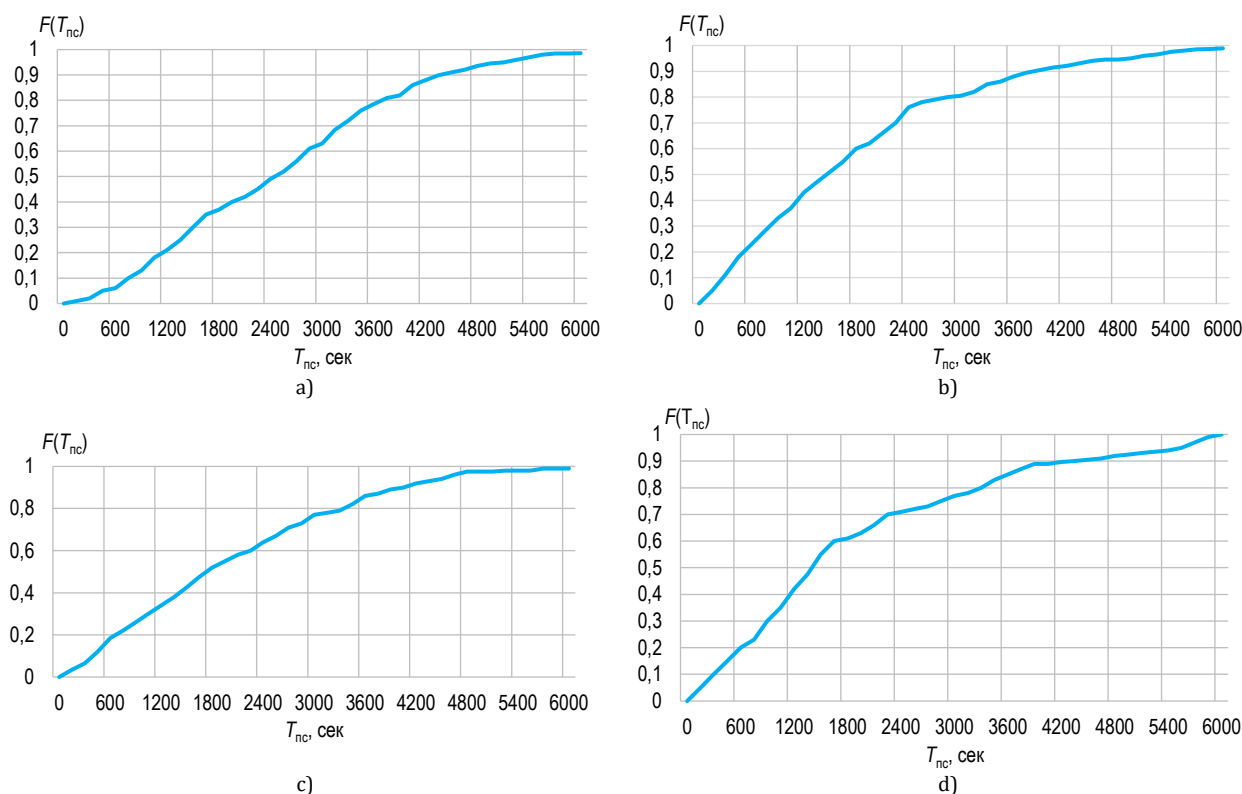


Рис. 2. Эмпирическая функция распределения случайной величины T_{nc} для ГНСС ГЛОНАСС (а), GPS (b), ГАЛИЛЕО (с) и КОМПАСС (d)

Fig. 2. Empirical Distribution Function of random Variable T_{nc} for GNSS GLONASS (a), GPS (b), GALILEO (c) and КОМПАСС (d)

никами, а также к спуфингу, который манипулирует местоположением и временем, вычисляемых приемниками. В условиях растущей зависимости от ГНСС для определения местоположения, навигации и синхронизации времени, чтобы избежать возможных последствий преднамеренных и непреднамеренных атак на сигналы, следует уделять особое внимание шагам по оценке и снижению уязвимости НАП ГНСС, а также технологиям защиты от помех и противодействия спуфингу.

Отмеченные виды представляют собой тип радиопомех ГНСС, которые возникают, когда слабые сигналы спутниковой системы подавляются более сильными радиосигналами на той же частоте, при этом спуфинг можно отождествлять как интеллектуальную форму вмешательства, во время работы которой расположенный поблизости радиопередатчик посылает ложные сигналы в целевой приемник.

В данной работе были проанализированы процессы поиска навигационных сигналов, которые определяются как задача оценки его задержки и ДСЧ, которые принимаются постоянными на интервале наблюдения, с целью определения характеристик уязвимости НАП ГНСС к спуфинг-атакам. С учетом ограничений и допущений в качестве показателя были выбраны время появления T_{nc} одного или нескольких навигационных спутников из состава орбитальной группировки из-за линии ра-

диогоризонта в зоне видимости НАП ГНСС и вероятность уязвимости к спуфинг-атаке P_{ca} .

Полученные путем моделирования эмпирические функции распределения позволили определить вероятность уязвимости к спуфинг-атаке навигационных приемников, работающих с сигналами различных ГНСС. Из результатов можно сделать вывод: воздействие спуфинг-атак снижает эффективность функционирования аппаратуры потребителя, не зависимо от типа спутниковой навигационной системы, с которой она сопряжена. Однако наилучшую устойчивость к спуфинг-атакам демонстрирует аппаратура, функционирующая по сигналам отечественной системы ГНСС ГЛОНАСС.

Полученные в данном эксперименте результаты могут быть использованы отечественными организациями-разработчиками при создании НАП ГНСС нового поколения с улучшенной защитой от спуфинга, а также в качестве дополнения к процессам управления уязвимостями НАП ГНСС на всем жизненном цикле системы менеджмента безопасности, включая этапы мониторинга и оценки их применимости, определения уровня критичности применительно к НАП ГНСС, определения методов и приоритетов устранения и принятия мер, направленных на устранение или исключение возможности использования (эксплуатации) выявленных уязвимостей.

Список источников

1. Лемешко Н., Патшин А. Оценка качества функционирования спутниковых навигаторов при наличии помех с использованием векторных генераторов R&S sMW200A и R&S sMBV100B // Электроника: наука, технология, бизнес. 2020. № 3. С. 122–125. DOI:10.22184/1992-4178.2020.194.3.122.125
2. Рубцов Н.С. Алгоритм защиты от спуффинга аппаратуры потребителей спутниковых навигационных систем // Известия Тульского государственного университета. Технические науки. 2018. № 4. С. 92–101.
3. Перова А. И., Харисова В.Н. ГЛОНАСС. Принципы построения и функционирования. М.: Радиотехника, 2010. 800 с.
4. Яценков В.С. Основы спутниковой навигации. Системы GPSNAVSTAR и ГЛОНАСС. М.: Горячая линия – Телеком, 2005. 272 с.
5. Лемешко Н., Патшин А. Оценка качества функционирования спутниковых навигаторов при наличии помех с использованием векторных генераторов R&S sMW200A и R&S sMBV100B Часть 2 // Электроника: наука, технология, бизнес. 2020. № 4. С. 78–84. DOI:10.22184/1992-4178.2020.195.4.78.84
6. Неровный В.В., Кирюшкин В.В., Бабусенко С.И., Коратаев П.Д., Облов П.С. Вероятностные характеристики системы поиска и обнаружения навигационных сигналов в условиях имитирующих помех // Радиотехника. 2023. Т. 87. № 7. С. 60–66. DOI:10.18127/j00338486-202307-07
7. Grewal M.S., Weill L.R., Andrews A.P. Global Positioning Systems, Inertial Navigation, and Integration. John Wiley & Sons, Inc., 2001. DOI:10.1002/0471200719
8. Советов Б.Я., Яковлев С.А. Моделирование систем: учеб. для вузов. М.: Высшая школа, 2001. 343 с.

References

1. Lemeshko N., Patshin A. Evaluation of Satellite Navigators Performance Under Interference Using R&S sMW200A and R&S sMBV100B Vector Generators. *Electronics: Science, Technology, Business*. 2020;3:122–125. DOI:10.22184/1992-4178.2020.194.3.122.125
2. Rubtsov N.S. Algorithm for protection of GNSS Receivers from Spoofing. *Izvestia Tula State University*. 2018;4:92–101.
3. Perova A.I., Kharisova V.N. *GLONASS. Principles of Construction and Functioning*. Moscow: Radiotekhnika Publ.; 2010. 800 p.
4. Yatsenkov V.S. *Basics of Satellite Navigation. GPSNAVSTAR and GLONASS Systems*. Moscow: Goryachaya liniya – Telekom Publ.; 2005. 272 p.
5. Lemeshko N., Patshin A. Evaluation of Satellite Navigators Performance Under Interference Using R&S sMW200A and R&S sMBV100B Vector Generators Part 2. *Electronics: Science, Technology, Business*. 2020;4:78–84. DOI:10.22184/1992-4178.2020.195.4.78.84
6. Neronov V.V., Kiryushkin V.V., Babusenko S.I., Korataev P.D., Oblov P.S. Probabilistic Characteristics of the Navigation Signal Search and Detection System Under Conditions of Simulated Interference. *Radioengineering*. 2023;87(7):60–66. DOI:10.18127/j00338486-202307-07
7. Grewal M.S., Weill L.R., Andrews A.P. *Global Positioning Systems, Inertial Navigation, and Integration*. John Wiley & Sons, Inc.; 2001. DOI:10.1002/0471200719
8. Sovetov B.Ya., Yakovlev S.A. *Modeling of systems*. Moscow: Vysshaya shkola Publ.; 2001. 343 p.

Статья поступила в редакцию 05.11.2023; одобрена после рецензирования 20.11.2023; принята к публикации 04.12.2023.

The article was submitted 05.11.2023; approved after reviewing 20.11.2023; accepted for publication 04.12.2023.

Информация об авторе:

НЕРОВНЫЙ
Валерий Владимирович

доктор технических наук, доцент, профессор кафедры приемных и передающих радиоустройств (средств и связи РТО) Военно-воздушной академии им. проф. Н.Е. Жуковского и Ю.А. Гагарина
 <https://orcid.org/0009-0006-2385-2718>

КОРАТАЕВ
Павел Дмитриевич

кандидат технических наук, старший преподаватель кафедры авиационных систем и комплексов радионавигации и радиосвязи Военно-воздушной академии им. проф. Н.Е. Жуковского и Ю.А. Гагарина
 <https://orcid.org/0009-0000-3634-2772>

ОБЛОВ
Пётр Сергеевич

адъюнкт кафедры приемных и передающих радиоустройств (средств и связи РТО) Военно-воздушной академии им. проф. Н.Е. Жуковского и Ю.А. Гагарина»
 <https://orcid.org/0009-0005-5885-6463>

ТОЛСТЫХ
Марина Юрьевна

кандидат технических наук, доцент кафедры международной информационной безопасности института информационных наук Московского государственного лингвистического университета, доцент кафедры специальных информационных технологий учебно-научного комплекса информационных технологий Московского университета МВД России имени В.Я. Кикотя
 <https://orcid.org/0009-0001-2223-7709>

РЕЗУЛЬТАТЫ ИССЛЕДОВАНИЙ МОЛОДЫХ УЧЕНЫХ

*Основой всей научной работы служит убеждение,
что мир представляет собой упорядоченную
и познаваемую сущность...*

Альберт Эйнштейн

1.2.2

2.2.6

2.2.13

2.2.14

2.2.15

2.2.16

2.3.1

2.3.6

Научная статья

УДК 004.5

DOI:10.31854/1813-324X-2023-9-6-102-115



Эффективная модель интерфейса взаимодействия пользователя с информационным сервисом запросного типа

Павел Андреевич Курта, expert@kurta.ru

ООО «Норд клининг»,
Мурманск, 183031, Российская Федерация

Аннотация: В работе решается задача построения аналитической модели интерфейса взаимодействия пользователя с информационной системой, предназначенной для выдачи информации на основании многопараметрических запросов пользователя. Назначением модели является оценка эффективности интерфейса и, соответственно, самого взаимодействия. Для этого, используя пример типовой работы с интерфейсом, вводятся следующие предпосылки (как условия) для моделирования: эффективность интерфейса, его формы и переходы между ними, элементы интерфейса, атомарная эффективность элемента и формы, порядок их расположения и эмпирическая зависимость эффективности, а также постановка задачи проектирования интерфейса. Используя указанные предпосылки, сформирована соответствующая эффективностная модель, позволяющая аналитически, на основании структуры и параметров интерфейса, оценивать его результативность, оперативность и ресурсоэкономность. Создан программный прототип, в основе которого лежит модель и с помощью которого проведен ряд экспериментов по трем сценариям, заключающимся в изменении количества форм и элементов интерфейса с целью определения их влияния на общую эффективность; получены соответствующие зависимости в виде гистограмм. Сделаны выводы касательно точности модели, ее теоретической и практической значимости, а также общей сложности проектирования «идеальных» интерфейсов.

Ключевые слова: информационная система, интерфейс, предпосылки, аналитическая модель, эффективность, прототип, эксперимент

Ссылка для цитирования: Курта П.А. Эффективная модель интерфейса взаимодействия пользователя с информационным сервисом запросного типа // Труды учебных заведений связи. 2023. Т. 9. № 6. С. 102–115. DOI:10.31854/1813-324X-2023-9-6-102-115

An Efficient Interface Model of User Interaction with a Query-Type Information Service

Kurta Pavel, expert@kurta.ru

Nord Cleaning Ltd,
Murmansk, 183031, Russian Federation

Abstract: The work solves the problem of an analytical model constructing of the user interaction interface with an information system designed to provide information based on multi-parameter user requests. The model purpose is to evaluate the effectiveness of the interface and, accordingly, the interaction itself. To do this, using an example of typical work with an interface, the following prerequisites (as conditions) for modeling are introduced: the effectiveness of the interface, its forms and transitions between them, interface elements, the atomic efficiency of the element and form, the order of their arrangement and the empirical dependence of efficiency, as well as the formulation of interface design tasks. Using these premises, an appropriate efficiency model has been formed, which allows us to analytically, based on the structure and parameters of the interface, evaluate its effectiveness, operativeness and resource

efficiency. A software prototype was created, which was based on the model and with the help of which a number of experiments were carried out under three scenarios, consisting of changing the number of forms and interface elements in order to determine their impact on overall efficiency; the corresponding dependencies were obtained in the form of histograms. Conclusions are drawn regarding the accuracy of the model, its theoretical and practical significance, as well as the overall complexity of designing "ideal" interfaces.

Keywords: information system, interface, prerequisites, analytical model, efficiency, prototype, experiment

For citation: Kurta P. An Efficient Interface Model of User Interaction with a Query-Type Information Service. *Proceedings of Telecommun. Univ.* 2023;9(6):102–115. DOI:10.31854/1813-324X-2023-9-6-102-115

Введение

Одним из преимуществ повсеместного внедрения информационных технологий стало предоставление человеку возможностей работы с большими объемами гетерогенной информации; например, внедрение концепции Умного дома в быту, обеспечение высокоскоростного доступа к информационным ресурсам практически из любой точки мира, предоставление справочных, обучающих и развлекательных сервисов, а также многое другое. В такой структуре взаимодействия человека с информационной системой (ИС) особое место занимают точки их соединения – интерфейсы (как некоторая совокупность и логика перехода между элементами двухсторонней передачи информации), «удачность» проектирования которых влияет на их эффективность, а также и само взаимодействие [1]. Здесь и далее под эффективностью понимается ее авторская трактовка [2], определяемая тремя классическими показателями: результативностью – отсутствием ошибок в данных, оперативностью – скоростью взаимодействия, и ресурсоэкономностью – сохранностью расходуемых ресурсов. При этом, на характеристики взаимодействия оказывает влияние не только состав интерфейсных элементов, но и состояние человека, особенности обмениваемых данных и ряд других, не всегда очевидных, вещей. Например, выбор человеком страны проживания, как правило, будет осуществлен быстрее и точнее через выпадающий список, чем ручной ввод; и наоборот, указание числа от 1 до 1000 быстрее путем непосредственного ввода с цифрового блока клавиатуры, нежели выбором из списка или «ползунком», хотя точность ввода в обоих случаях будет одинаковой. Также, находящемуся в состоянии усталости человеку комфортнее будет обрабатывать информацию, выводимую порциями – на отдельных экранных формах, а не всю сразу; если же, требуется оперативный анализ больших наборов данных в рамках должностных обязанностей, то целесообразнее предоставлять информацию оператору моментально, а не через последовательно появляющиеся формы. Тем не менее, преобладающее большинство интерфейсов современных программно-аппаратных средств (программ на персональных компьютерах, банкоматах, информационно-электронных

стендах в торговых центрах и т. п.) учитывают лишь общие требования к эргономике интерфейсов, дизайнерские решения отдельных разработчиков или частные методики сравнения и выбора вариантов (как при А/В-тестировании) без рассмотрения остальных участников процесса взаимодействия: заказчика информации – человека, и носителя информации – данных.

Приведенные выше примеры и выводы «очерчивают» предметную область текущего исследования, как – *интерфейсы взаимодействия человека с ИС с позиции их эффективности*. В связи с этим возникает следующее научное противоречие предметной области, сформулированное в виде противопоставления возможностей (т. е. имеющегося состояния дел) и потребностей (т. е. необходимого состояния дел). С одной стороны, используемые и разрабатываемые интерфейсы ИС строятся по достаточно абстрактным схемам (отстраненным от реальности), не учитывающим внешние и внутренние стороны процесса обмена информацией (пользователей, данные, условия применения и т. п.). С другой стороны, одним из основных факторов повышения эффективности интерфейса является учет восприятия информации пользователем [3], на которое влияет, как указывалось, его состояние (психическое, физиологическое и иное), способ ввода и предоставления данных, а также порядок такого предоставления и т. п.

Первым же существенным шагом по разрешению противоречия может стать построение аналитической модели такого взаимодействия (далее – Модель), позволяющей формально оценивать его эффективность – т. е. *эффективностной модели*, что и является задачей текущего исследования [4]. А исходя из того, что при существующем разнообразии целей и задач ИС, создание для них единой Модели является целой проблемой, то ограничимся следующим их подклассом (согласно авторской таксономии). Во-первых, будем рассматривать только ИС, являющиеся информационными сервисами [5] – т. е. ориентированными на решение определенной задачи и обладающими алгоритмами для ее решения. Во-вторых, возьмем лишь информационные сервисы запросного типа – т. е. те, в которых инициатором взаимодействия является человек (например, справочные системы, осу-

ществляющие поиск и предоставление информации по ключевым словам); данный подкласс ИС будем далее сокращенно называть – Сервисом. В-третьих, не будем учитывать функционал интерфейсов, заключающийся в зависимости порядка отображения и состава новых форм на основании ввода значений на предыдущих (исключив тем самым, например, системы с оплатой услуг несколькими способами: банковскими картами с необходимостью ввода всех ее реквизитов, по QR-коду с отображением изображения для сканирования или через систему быстрых платежей с полем ввода номера мобильного телефона) – т. е. логику переходов между ними будем считать линейной, что тем не менее позволит смоделировать огромное количество реально функционирующих ИС. И, в-четвертых, поскольку основным предназначением Сервиса является итоговая выдача запрошенных данных, элементом вывода будем считать только текстовое поле (с его помощью возможно отобразить любую информацию, даже 3D-изображение). Отметим, что помимо данного типа Сервиса, автором выделяется еще два следующих, не рассматриваемых далее: управляющий – который сам является инициатором (например, устройства Умного дома, требующие реакции от человека) и игровой – в котором инициатором взаимодействия являются оба участника (например, любой игровой автомат).

Предпосылки к созданию

Для построения Модели применим следующий *эмпирико-индуктивно-формализационный* методологический подход. Суть его будет заключаться в том, что изначально предложим пример типовой работы пользователя с интерфейсом Сервиса – т. е. будем основываться на данных, полученных исходя из гипотетического эксперимента. Затем по его результатам предложим базовые предпосылки для создания Модели – будет осуществлен переход от частного к общему. И, наконец, систематизируем предпосылки в единую схему работы интерфейса, которая затем подвергнется формализации. А поскольку назначением Модели является оценка эффективности интерфейса, то в ней должны будут отражены механизмы такой оценки.

Предположим, что Сервисом является поиск телефонного номера человека в базе данных по его полному имени, дате рождения и месту проживания. И хотя вариаций интерфейсов даже для такой типовой задачи может быть достаточно много (все элементы расположатся на одной интерфейсной форме или же для каждого элемента используется своя форма и при этом ввод даты рождения можно осуществлять как численно, так и через встроенный календарь), выберем на взгляд автора наиболее гармоничный интерфейс. Для ввода будут использоваться три формы, каждая из которых предназначена для своей группы данных – 1)

имя, фамилия и отчество (ФИО); 2) день, месяц и год рождения; 3) страна и город проживания; соответственно, на каждой форме, кроме последней, должна присутствовать кнопка продолжения работы (для перехода на следующую). В первой группе данных будем использовать текстовый ввод ФИО, во второй – выбор из выпадающего списка соответствующих чисел (дня – от 1 до 31, месяца – от 1 до 12 и года – от 1900 до 2023), а в третьей – выбор из выпадающего списка страны и ввод в текстовое поле города. Для вывода найденной информации будет предназначаться форма с одним текстовым полем в формате «+X (XXX) XXX – XX – XX», где X – цифра телефонного номера.

Процесс единичного решения задачи человека с помощью Сервиса описывается следующей последовательностью действий. Пользователю открывается 1-я форма, элементы которой он должен заполнить. В первый момент пользователем затрачивается определенное количество времени и умственной деятельности на восприятии формы, зависящее, в частности, от общего количества элементов на ней; затем пользователь начинает процесс ввода данных. Из-за необходимости указывать ФИО вручную, с одной стороны, повышается риск ошибки во введенных данных, а с другой – уменьшается время ввода по сравнению с тем, если бы пришлось выбирать каждый компонент (фамилию, имя и отчество) из выпадающего списка (из-за их огромного многообразия).

Оценка психоэмоционального напряжения (ПЭН), вызванного затратами ресурсов человека в случае ручного ввода или выбора из списка, является предметом дискуссии и требует проведения дополнительных исследований; однако очевидно, что оно будет различным. После окончания работы с 1-й формой пользователь должен найти кнопку перехода на 2-ю форму и нажать на нее (некие временные издержки и затрата «нервов»). На этой форме пользователь производит ввод даты рождения с меньшим количеством ошибок, по сравнению с вводом ФИО, хотя расходует больше времени. При этом как по мере перехода к элементам в рамках формы, так и при продвижении к следующим формам все показатели эффективности будут незначительны, но снижаться в связи с нарастанием ПЭН пользователя (поскольку ему необходимо воспринять «картинку», выделить ее компоненты, проанализировать и проинтерпретировать новый вид и суть представляемой информации). Все это будет вести к увеличению количества ошибок при вводе, замедлению работы с элементами, а также ухудшению психоэмоциональное состояние.

Работа с 3-й формой будет сочетать как выбор из выпадающего списка одних значений (страны), так и текстовый ввод других (города), в результате чего предсказание показателей эффективности

становится еще более сложным. На 4-й форме отображается итоговая информация, состоящая из одного значения, вариация чего в общем представляется маловероятной.

Отметим, что конструкция интерфейса, в которой осуществляется переход к новым формам с постепенным заполнением полей, достаточно популярна. Причина этого, вероятно, заключается в субъективном удобстве интерфейса для большинства пользователей. В качестве примера интерфейса можно привести Интернет-сервис по подаче деклараций о налоговом вычете на Web-сайте Федеральной Налоговой Службы (рисунок 1).

Согласно проведенному гипотетическому эксперименту по использованию простого интерфейса поискового Сервиса, эффективность работы пользователя с каждым элементом была различной – для части из них возрастала вероятность ошибок, на другие затрачивалось большее количество времени, а после третьих у пользователя нарастало ПЭН. Сами данные, передаваемые через интерфейс, также оказывали влияние на эффективность – в зависимости от используемых для них элементов. Каждая форма при первом появлении также в той или иной степени затрачивала человеческие ресурсы на осознание представленной на ней информации. При этом разбиение всего интерфейса на множество форм приводило как к появлению дополнительных элементов – кнопок, так и влияло на общее ПЭН от работы (не обязательно в сторону роста напряжения).

Важнейшим выводом из эксперимента является тот факт, что конструирование «идеального» – в смысле максимально эффективного – интерфейса представляется крайне сложно решаемой задачей даже теоретически, поскольку повышение одного показателя эффективности (например, результативности путем замены текстового поля на выпадающий список) в большинстве случаев ведет к снижению других (например, оперативности из-за необходимости выбора из огромного списка нужного элемента и ресурсоэкономности из-за увеличения ПЭН пользователя при таком «скроллинге» списка) [6]. Ситуация усложняется тем, что более дальние формы (с позиции «дальности» на пути продвижения по интерфейсу) и элементы (с позиции «дальности» расположения на форме) начинают снижать показатели эффективности из-за особенностей функционирования человека (в отличие, например, от автомата), что приводит к необходимости учета структуры интерфейса – с позиции его разбиения на формы и размещения на них элементов, и параметров интерфейса – с позиции выбора типов элементов, передаваемых данных и т. п. При этом, большее число форм с одной стороны добавит новые элементы, а с другой – позволит пользователю быстрее подготовиться к работе с ней, сохранив больше ресурсов. Все это говорит об общей сложности проектирования интерфейсов и необходимости в ряде случаев увеличивать не всю эффективность, а лишь отдельные ее показатели – согласно заданным целям и условиям.

Рис. 1. Пример интерфейса Интернет-сайта Федеральной Налоговой Службы

Fig. 1. Example of the Internet Site of the Federal Tax Service Interface

Целесообразным же решением может стать подстройка интерфейса в режиме реального времени – путем указания требований к показателям для конкретных сфер деятельности, задач или групп пользователей [7]. Так, например, в банковской сфере, где в приоритете является точность, интерфейс можно спроектировать при максимизации его результативности; в сфере реагирования на чрезвычайные ситуации – стремясь увеличить оперативность; в сфере социальных услуг (особенно, для пожилых групп пользователей) – повышая ресурсоэкономность; а при размещении Сервисов в общедоступных местах – позволяя настраивать показатели самостоятельно. Касательно последнего можно привести довольно интересную гипотетическую ситуацию, когда пользователь ИС в торговом центре на основании своего текущего состояния (усталости, спешки, знакомства с сервисом и т. п.) сможет точно подстроить интерфейс для работы.

Систематизируем частные выводы из гипотетического эксперимента путем их обобщения в виде предпосылок для построения Модели интерфейсов Сервиса данного – запросного – типа.

Предпосылка 1. Эффективность интерфейса

Эффективность интерфейса представляет собой совокупность трех классических показателей со следующей интерпретацией:

1) ресурсоемкость – мера допущения ошибок при вводе данных (поскольку их вывод при отсутствии дефектов в реализации интерфейса будет всегда корректным);

2) оперативность – быстрота обмена данными между человеком и Сервисом (в основном с точки зрения их ввода);

3) ресурсоэкономность – отсутствие или мера незначительности ПЭН человека при работе с интерфейсом (ресурсы Сервиса считаются несущественными, поскольку обмен и обработка сверхбольших объемов данных через формы или их элементы, требующих запредельных показателей аппаратного обеспечения ИС, выходят за рамки решения типовых задач Сервиса по реагированию на запросы).

Предпосылка 2. Формы интерфейса

Интерфейс состоит из последовательного перехода между формами, содержащими набор элементов; например, таким образом можно группировать ввод ФИО человека, данных о дате и месте рождения, городе проживания, семейном положении, образовании, интересах и т. п.

Предпосылка 3. Переход между формами

Движение по логике интерфейса осуществляется последовательно через все формы с помощью нажатия на соответствующей кнопки; например, в правом нижнем углу с надписью «Далее».

Предпосылка 4. Элементы интерфейса

На каждой форме присутствует последовательность элементов интерфейса для ввода или вывода; например, поля ФИО, как правило, располагаются вертикально друг за другом. Элементы могут типизироваться согласно их предназначению – для ввода или вывода информации, а также принадлежать к некоторому классу, определяющему то, как эта информация передается [8]. Далее будем рассматривать лишь следующие основные классы элементов ввода: текстовое поле, выпадающий список, флаговая кнопка (т. е. CheckBox, *транслит. на русс.* Чекбокс), двунаправленный счетчик и «ползунок»; элемент вывода будет представлен только текстовым полем, естественно, без возможности редактирования.

Предпосылка 5. Атомарная эффективность элемента

Каждый интерфейсный элемент определенного класса обладает собственной атомарной эффективностью взаимодействия с пользователем, зависящей (в грубом, но приемлемом приближении) от типа и размера передаваемых данных [9]. Далее будем рассматривать только наиболее популярные типы, а именно – булевские (состояние из двух значений: «1 и 0», «Да и Нет», «Правда и Ложь», и т. п.), числовые неотрицательные (все значения от 0 до условно большого целого числа) и текстовые (последовательности букв некоторого алфавита).

Предпосылка 6. Атомарная эффективность формы

Каждая интерфейсная форма обладает собственной атомарной эффективностью, определяемой потребностями пользователя в следующих действиях (в аналогичном для Предпосылки 5 приближении): вначале, восприятием отображаемых на форме элементов, на что влияет количество последних; в конце, нажатием на кнопку перехода на следующую форму. Заметим, что влияние кнопки перехода на показатели эффективности можно считать инвариантом для конкретной реализации интерфейса.

Предпосылка 7. Порядок элемента на форме

Показатели атомарной эффективности каждого элемента корректируются в зависимости от дальности расположения на интерфейсной форме – поскольку восприятие более поздних элементов отличается от восприятия более ранних; предположительно, оно ухудшается, хотя формально стоит учитывать любую зависимость.

Предпосылка 8. Порядок формы в интерфейсе

Показатели атомарной эффективности самой формы и всех ее элементов корректируются в зависимости от дальности ее расположения на всем пути логики интерфейса по аналогичным с Предпосылкой 7 причинам.

Предпосылка 9. Эмпирическая зависимость эффективности от порядка

Зависимость эффективности элементов и форм от их порядка представляется сложно определенной теоретическими методами и, поэтому, может быть получена эмпирически (например, с помощью экспериментов по сбору статистических данных на целевых группах пользователей).

Предпосылка 10. Задача проектирования интерфейса

Задача проектирования интерфейса сводится к достижению определенных значений (максимальных или конкретных) для одного или нескольких показателей эффективности путем его структурно-параметрического синтеза (комбинирования разбиения на формы, выбора состава и порядка расположения на них элементов, используемых классов, типов и размеров данных и т. п.); при этом вопросы уязвимости интерфейсов (т. е., например, изначальные ошибки в логике работы Сервиса) [10] в данном контексте не рассматриваются, поскольку выходят за рамки предметной области.

Формализация полученных предпосылок, которые, как хорошо видно, находятся в тесной логической связи, как раз и позволит создать требуемую Модель.

Аналитическая модель

Проведем пошаговую формализацию всех 10 указанных предпосылок к созданию Модели. В качестве обеспечения подобию будем придерживаться следующего стиля записей: нижний индекс – порядковый номер объекта в списке (например, форм); верхний индекс – идентификатор объекта для заранее известного или заданного множества (например, классов); « $\sim$ » – диакритический знак «волна» для выделения функций, вычисляющих результирующие значения по ее параметрам (например, атомарную эффективность по классу элемента, типу и размеру данных); « $\bar{}$ » – диакритический знак «две линии» для выделения процессов, продуцирующих функции (например, атомарную эффективность элемента); «'» и «''» – диакритические знаки «апостроф» и «двойной апостроф» для указания первоначально и вторично уточненного значения (т. е. после применения корректировки к основному и первоначально скорректированному); «:=» – символ определения сложной функциональной сущности (например, функции или задачи).

Предпосылка 1. Эффективность интерфейса.

Аналитическая запись

Эффективность интерфейса (*Efficiency*) имеет следующую запись:

$$\begin{aligned} Efficiency &\equiv \\ &\equiv \langle Potency, Operativeness, ResourceSaving \rangle, \end{aligned}$$

где « $\langle \rangle$ » – кортеж из последовательности следующих показателей эффективности: *Potency* – результативность, *Operativeness* – оперативность, *ResourceSaving* – ресурсоэкономность.

Как будет показано ниже (при расчете комплексной эффективности всего интерфейса), в качестве значений показателей целесообразнее брать некоторые меры в диапазоне $[0,1]$:

$$\begin{cases} Potency \in [0,1] \\ Operativeness \in [0,1] \\ ResourceSaving \in [0,1] \end{cases}.$$

С точки зрения результативности такой диапазон значений является достаточно логичным, поскольку данный показатель определяет меру допущения ошибок. Так, если какой-либо интерфейсный элемент принципиально не позволяет ввести данные (например, для ввода текста используется счетчик чисел), то его результативность всегда будет равна 0; а если для ввода единственного числа выбрана установленная и нередактируемая флаговая кнопка, то результативность будет равна 1 (что, впрочем, не имеет практического смысла). Саму результативность можно оценивать через действия (*Actions*), как отношение количества верных вводов (а формально, и выводов) данных ($Actions^{Valid}$) к количеству всех вводов ($Actions^{All}$):

$$Potency = \frac{Actions^{Valid}}{Actions^{All}}.$$

Подобная ситуация характерна и для ресурсоэкономности – максимально простая работа с элементом или формой (т. е. минимизация ПЭН) делает значение показателя равным 1, а максимальная сложная – 0. Саму ресурсоэкономность можно оценивать по результатам опроса пользователя (т. е. ответа на вопрос «Насколько сложно было работать?») по окончании единичного взаимодействия на основании его собственных ощущений (*Feeling*):

$$\begin{cases} ResourceSaving = \frac{Feeling^{Current}}{Feeling^{Max}} \\ Feeling^{Current} \in [0, Feeling^{Max}] \end{cases},$$

где $Feeling^{Current}$ – оценка пользователем своего текущего состояния по шкале, где 0 – психоэмоционально максимально затрудненная работа (вариант ответа в опроснике – «Невозможно»); $Feeling^{Max}$ – максимально простая и комфортная (вариант ответа в опроснике – «Элементарно»).

В случае оперативности, определение реальных значений показателя будет несколько иным, поскольку, как правило, быстрота процесса (или скорость) определяется временем работы, на которое делится ее объем (пройденное расстояние, количество движений или просто условная единица). При

таким подходе краевые значения оперативности достижимы лишь в пределе: 0 – в случае бесконечного времени работы с элементом, и бесконечность (формально получаемая при делении на близкое к нулю время) – при моментальном выполнении всех действий.

Привести данный показатель к единичной шкале можно с помощью времени работы ($Time$), которое является минимально возможным ($Time^{Min}$), определяемым эмпирическим путем, и которое будет делиться на текущее затраченное время ($Time^{Current}$):

$$Operativeness = \frac{Time^{Min}}{Time^{Current}}.$$

Предпосылка 2. Формы интерфейса. Аналитическая запись

Полный интерфейс Сервиса ($Interface$) имеет следующую запись:

$$Interface = [Form_1, \dots, Form_i, \dots, Form_N],$$

где «[]» – последовательность форм интерфейса; $Form_i$ – i -я форма; N – количество всех форм.

Поскольку каждая форма состоит из элементов ($Element$), то она имеет следующую запись:

$$Form = [Element_1, \dots, Element_j, \dots, Element_M],$$

где «[]» – последовательность элементов формы; $Element_j$ – j -й элемент; M – количество всех элементов на форме.

Предпосылка 3. Переход между формами. Аналитическая запись

Движения по интерфейсу через формы осуществляются с помощью интерфейсной кнопки перехода ($Button$), имеющей следующую запись:

$$Form_x^{Button} = Button_{x \rightarrow y},$$

где $Form_x^{Button}$ – кнопка на x -й форме; y – индекс формы, на которую по нажатию кнопки происходит переход с x -й; « $x \rightarrow y$ » – обозначение такого перехода между формами.

В условиях ограничений касательно линейности логики переходов между формами, очевидно, что интерфейс состоит из множества следующих кнопок перехода:

$$\begin{aligned} Interface^{Buttons} &= \\ &= \{Button_{1 \rightarrow 2}, \dots, Button_{x \rightarrow x+1}, \dots, Button_{N-1 \rightarrow N}\}, \end{aligned}$$

где «{ }» – множество кнопок.

Данное выражение верно, т. к. с каждой x -й формы возможен переход только на $x + 1$ -ю.

Поскольку последняя форма с выводом результата, очевидно, не содержит кнопки перехода, то количество физических кнопок будет на один меньше количества форм в интерфейсе; для сохранения же подобности и гармоничности формальной записи можно считать, что отсутствие кнопки на последней форме сопоставимо с виртуальной кнопкой $Button_{N \rightarrow N+1}$, которая не влияет на эффективность, т. е. ее эффективность:

$$Efficiency^{Button_{N \rightarrow N+1}} = \{1, 1, 1\}.$$

Отметим, что кнопка завершения работы на последней форме отсутствует, поскольку предполагается, что взаимодействие с Сервисом заканчивается в момент вывода итогового результата.

Предпосылка 4. Элементы интерфейса. Аналитическая запись

Тип элемента интерфейса ($Element^{Type}$) имеет следующую запись, соответствующую способности осуществлять двухсторонний обмен между пользователем и Сервисом:

$$Element^{Type} \in \{Element^{\rightarrow}, Element^{\leftarrow}\},$$

где «{ }» – множество типов элементов; $Element^{\rightarrow}$ и $Element^{\leftarrow}$ – тип элемента, предназначенный для ввода и вывода данных, соответственно (символ верхнего индекса-стрелки выбран, исходя из направления передачи данных – от пользователя к Сервису).

Также элементы ввода могут принадлежать различным классам (являющимся подмножеством типа), где $Element^{Class \rightarrow}$ – класс элемента ввода данных; а другие верхние индексы для $Element$ соответствуют классам следующим образом: $TextBox$ – текстовое поле; $DropDownList$ – выпадающий список; $CheckBox$ – флаговая кнопка; $Spinner$ – двунаправленный счетчик; $Slider$ – «ползунок» (1).

Согласно указанным ограничениям, элементы вывода могут относиться только к одному классу:

$$Element^{Class \leftarrow} \in \{Element^{TextBoxRO}\},$$

где верхний индекс $TextBoxRO$ соответствует текстовому полю без возможности редактирования (постфикс RO – аббр. от англ. *Read Only*; пер. на русс. Только чтение).

Исходя из того, что множества классов элементов для различных направлений передачи данных не пересекаются, их можно объединить в один класс, опуская тем самым (здесь и далее) указание типа элемента:

$$Element^{Class} = Element^{Class \rightarrow} \cup Element^{Class \leftarrow},$$

где $\cup$ – операция объединения множеств классов элементов.

$$Element^{Class \rightarrow} \in \{Element^{TextBox}, Element^{DropDownList}, Element^{CheckBox}, Element^{Spinner}, Element^{Slider}\}. \quad (1)$$

Предпосылка 5. Атомарная эффективность элемента. Аналитическая запись

Атомарная эффективность интерфейсного элемента ($Efficiency_{Element_j}$) имеет вид (2), где $Efficiency^{Element}(\dots)$ – функция вычисления атомарной эффективности элемента согласно его классу ($Element^{Class}$) и свойствам данных ($Data$), состоящим из типа ($Data^{Type}$) и размера ($Data^{Size}$), для передачи которых он используется.

Тип данных записывается следующим образом:

$$Data^{Type} \in \{Data^{Boolean}, Data^{Unsigned}, Data^{Text}\},$$

где « $\{ \}$ » – множество типов данных; а индексы для $Data$ соответствуют типам следующим образом: *Boolean* – булевский; *Unsigned* – целое беззнаковое число; *Text* – текст.

Предпосылка 6. Атомарная эффективность формы. Аналитическая запись

Атомарная эффективность интерфейсной формы ($Efficiency_{Form}$) имеет следующую запись:

$$Efficiency_{Form} = Efficiency^{Form}(M_{Form}, Button_{x \rightarrow x+1}),$$

где $Efficiency^{Form}(\dots)$ – функция вычисления атомарной эффективности формы согласно количеству элементов на ней (M_{Form}) и кнопке перехода на следующую форму (для x -й формы – $Button_{x \rightarrow x+1}$).

Исходя из того, что влияние кнопки на показатели эффективности считается инвариантом в рамках одного интерфейса, будет верно следующее уравнение (для любой x -ой формы с одинаковым количеством элементов):

$$Efficiency^{Form}(M, Button_{x-1 \rightarrow x}) = Efficiency^{Form}(M, Button_{x \rightarrow x+1}),$$

т. е. атомарные эффективности форм будут отличаться только при разном количестве элементов на них; в ином случае, следуя уравнению, они равны.

Предпосылка 7. Порядок элемента на форме. Аналитическая запись

Эффективность интерфейсного элемента после коррекции ($Efficiency'_{Element}$), которая учитывает порядок его расположения на форме, имеет вид (3), где $EfficiencyCorrection^{Element}(\dots)$ – функция вычисления скорректированной эффективности элемента по ее атомарному значению и порядковому индексу (*Index*).

Предпосылка 8. Порядок формы в интерфейсе. Аналитическая запись

Эффективность элемента ($Efficiency''_{Element}$) и формы ($Efficiency'_{Form}$) после коррекции, которая учитывает порядок формы в логике интерфейса, имеет вид (4), с использованием единой функции.

В выражении (4) $EfficiencyCorrection^{Form}(\dots)$ – функция вычисления скорректированных эффективностей элемента и формы по первоначально скорректированной (согласно Предпосылке 7) и атомарной, соответственно; *Index* – порядковый индекс формы в логике интерфейса, согласно которому осуществляется коррекция. Таким образом, эффективность формы подвергается однократной корректировке по порядковому индексу, а эффективность элемента – двухкратной корректировке по паре порядковых индексов.

Предпосылка 9. Эмпирическая зависимость эффективности от порядка. Аналитическая запись

Поскольку каждая из введенных функций определяет достаточно сложную зависимость между своими параметрами и выходным значением (например, оперативность работы пользователя с двунаправленным списком вряд ли будет линейно зависеть от количества отображаемых им значений), то способ определения такой взаимосвязи может быть определен статистически, путем проведения соответствующих экспериментов. Всего насчитываются 4 их группы, в рамках каждой из которых определяются следующие зависимости: 1) атомарная эффективность элемента от комбинации его класса, а также типов и размеров данных (см. Предпосылку 5); 2) атомарная эффективность формы от количества элементов на ней и различных кнопок перехода (см. Предпосылку 6); 3) скорректированная эффективность элемента от его порядка на форме (см. Предпосылку 7); 3) скорректированная эффективность формы от ее порядка в логике интерфейса (см. Предпосылку 8). Данные группы экспериментов (*Experiment*) могут быть записаны в формальном виде (5), где $Experiment$ – процесс получения функции (т. е. алгоритма вычисления значений по параметрам согласно некоторым их соотношениям) для вычисления или корректировки эффективностей, верхний индекс которого определяет назначение функции: *Element* – атомарная эффективность элемента, *Form* – атомарная эффективность формы, *ElementCorrection* – корректировка атомарной эффективности элемента, *FormCorrection* – корректировка атомарной эффективности формы; $\{Subject_i\}$ – множество i -ых субъектов (участников эксперимента), согласно процессу взаимодействия с интерфейсом которых вычисляются показатели эффективности; $\{Element_i^{Class}\}$ – множество i -ых классов элементов (в количестве шести); $\{Date_i^{Type}\}$ – множество i -ых типов данных (в количестве трех); $\{Date_i^{Size}\}$ – множество i -ых размеров данных (например, от 1 до выбранного разумного предела с увеличением значения в 10 раз); $\{M_i\}$ – множество i -ых значений количества элементов на форме (также, от 1 до выбранного разумного предела с шагом 1); $\{Button_i\}$ –

множество i -ых вариаций кнопки перехода (без потери точности достаточно взять один вариант, стандартный для операционной системы интерфейса); $\{Index_j\}$ и $\{Index_i\}$ – множество j -ых индексов положения элемента на типовой форме и i -ых положения формы в логике интерфейса (например, от 1 до некоторого разумного предела с шагом 1).

Результаты экспериментов могут иметь вид зависимостей (аналитической или табличной), которые будут служить основой для вычисления соответствующих функций; например, если зависимость атомарной эффективности формы от количества элементов на ней имеет логистический вид, то в результате проведения эксперимента будут получен соответствующий набор точек на плоскости, сглаженная линия по которым и ляжет в основу вычислений соответствующей функции.

Каждый же эксперимент (*Experiment*, т. е. без указания верхнего индекса для назначения итоговой функции), очевидно, для одного набора параметров позволит получить сразу 3 зависимости (*Dependence*) – для всех показателей эффективности, где $Dependence^{Potency}$, $Dependence^{Operativeness}$ и $Dependence^{ResourceSaving}$ – зависимость от параметров эксперимента (и, следовательно, функции) результативности, оперативности и ресурсоэкономности соответственно (6).

Предпосылка 10. Задача проектирования интерфейса. Аналитическая запись

Общая эффективность для интерфейса, содержащего определенную структуру и параметры форм, кнопок перехода, элементов и их данных, имеет вид (7), где индекс оператора умножения i от 1 до N соответствует перебору всех форм интерфейса;

индекс j от 1 до M_i – перебору всех элементов на i -й форме с их количество M_i ; пара « i, j » – комплексному индексу, обозначающему j -й элемент (и его данные) на i -й форме.

Суть определения общей эффективности состоит в перемножении двух следующих выражений. В первом множителе производится вычисление атомарных эффективностей всех форм интерфейса с применением функции их корректировки согласно порядку формы. Во-втором множителе производится вычисление атомарной эффективности каждого элемента, расположенного на каждой форме, согласно его классу, типу данных и их размеру. Затем, эффективность элемента корректируется с учетом его порядка на каждой форме, а также порядка самой формы в логике интерфейса. Исходя из этих выражений, выбирая разбиение элементов по формам, классы элементов, учитывая передаваемые данные, изменяя порядок обоих сущностей, а также упорядочивая формы и элементы на них, можно «играться» итоговыми показателями эффективности – т. е. производить ее многокритериальную оптимизацию.

Необходимо указать, что умножение атомарных, скорректированных, общих и иных эффективностей заключается в поэлементном умножении ее показателей, т. е.:

$$\begin{aligned} Efficiency^X \times Efficiency^Y &\equiv \\ &Potency^X \times Potency^Y, \\ &\equiv \langle Operativeness^X \times Operativeness^Y, \rangle, \\ &ResourceSaving^X \times ResourceSaving^Y \end{aligned}$$

где x и y – верхние индексы, относящиеся к перемножаемым эффективностям.

$$\left\{ \begin{aligned} Efficiency_{Element} &= Efficiency_{Element}^{Element} (Element^{Class}, Data), \\ Data &\equiv \langle Data^{Type}, Data^{Size} \rangle \end{aligned} \right. \quad (2)$$

$$Efficiency'_{Element} = Efficiency_{Correction}^{Element} (Efficiency_{Element}, Index), \quad (3)$$

$$\left\{ \begin{aligned} Efficiency''_{Element} &= Efficiency_{Correction}^{Form} (Efficiency'_{Element}, Index) \\ Efficiency'_{Form} &= Efficiency_{Correction}^{Form} (Efficiency_{Form}, Index) \end{aligned} \right. \quad (4)$$

$$\left\{ \begin{aligned} Efficiency^{Element} &:= \overline{Experiment}^{Element} (\{Subject_i\}, \{Element_i^{Class}\}, \{Date_i^{Type}\}, \{Date_i^{Size}\}) \\ Efficiency^{Form} &:= \overline{Experiment}^{Form} (\{Subject_i\}, \{M_i\}, \{Button_i\}) \\ Efficiency_{Correction}^{Element} &:= \overline{Experiment}^{ElementCorrection} (\{Subject_i\}, \{Index_j\}) \\ Efficiency_{Correction}^{Form} &:= \overline{Experiment}^{FormCorrection} (\{Subject_i\}, \{Index_j\}) \end{aligned} \right. \quad (5)$$

$$\langle Dependence^{Potency}, Dependence^{Operativeness}, Dependence^{ResourceSaving} \rangle = \overline{Experiment} (...), \quad (6)$$

$$\begin{aligned} Efficiency^{Interface} &= \prod_{i=1}^N Efficiency_{Correction}^{Form} (Efficiency_{Form}^{Form} (M_i, Button_{i \rightarrow i+1}), i) \times \\ &\times \prod_{i=1}^N \prod_{j=1}^{M_i} Efficiency_{Correction}^{Form} (Efficiency_{Correction}^{Element} (\\ &Efficiency_{Element}^{Element} (Element_{i,j}^{Class}, \langle Data^{Type}, Data^{Size} \rangle_{i,j}), j), i) \end{aligned} \quad (7)$$

Обоснованность применения умножений (вместо типичного суммирования) при расчете общей эффективности заключается в том, что, как указывалось, показатели отражают степень достижения максимальных нормированных значений (т. е. единицы), а каждый элемент, форма или кнопка эти значения могут уменьшать. Например, если на форме присутствует 3 элемента, ухудшающих безошибочность ввода данных на 10 %, то результативность каждого равна 0.9, а результативность всей группы элементов в случае суммирования будет $0.9 + 0.9 + 0.9 = 2.7$, что, очевидно, неверно; правильным будет перемножение результативностей, т. е. что даст значение показателя – $0.9 \times 0.9 \times 0.9 = 0.729$.

Таким образом, общую задачу проектирования интерфейсов (*Task*), исходя из их целевой (*Target*) эффективности, можно записать в виде (8), где *Interface*() – интерфейс, спроектированный на

основании множества *i*-ых форм $\{Form_i\}$ (естественно, с нужным количеством кнопок перехода) и множества *j*-ых элементов на каждой *i*-ой форме $\{Element_{i,j}\}$, каждый из которых имеет соответствующий класс $Element_{i,j}^{class}$; « $\Rightarrow$ » – суть задачи в виде указания искомым параметрам (слева), в результате которых должны выполняться условия (справа); « $\rightarrow$ » – оператор стремления (в пределе, равенства) эффективности к целевому значению (*Target*), состоящему из совокупности 3-х показателей – $Target^{Operativeness}$, $Target^{ResourceSaving}$ и $Target^{Potency}$, соответствующих показателям результативности, оперативности и ресурсоэкономности; $Target^{Value}$ – значение, которое может принимать показатель целевого значения эффективности и которое может быть следующим: *Certain* – точное число, *Max* – максимально допустимое число, *Any* – произвольное число (означающее, что при решении задачи данный показатель не должен учитываться).

$$\left\{ \begin{array}{l} Task := \\ Interface(\{Form_i\}, \{Element_{i,j}\}, \{Element_{i,j}^{class}\}) \Rightarrow \\ Efficiency^{Interface} \rightarrow Target \\ Target \equiv \langle Target^{Potency}, Target^{Operativeness}, Target^{ResourceSaving} \rangle \\ Target^{Value} \in \{Certain, Max, Any\} \end{array} \right. \quad (8)$$

Так, решение задачи проектирования интерфейса для банковской сферы, когда важна лишь точность, может осуществляться при следующих целевых значениях показателей:

$$Target \rightarrow \langle Max, Any, Any \rangle.$$

При реагировании на чрезвычайные ситуации, требующем высокой оперативности [11], целевые показатели будут следующими:

$$Target \rightarrow \langle Any, Max, Any \rangle.$$

А в сфере социальных услуг для пожилых групп [12] целевыми показателями могут быть заданы, как:

$$Target \rightarrow \langle Any, Any, Max \rangle,$$

поскольку тут уже важна ресурсоэкономность. Если же при этом необходима и точность передачи данных от посетителей, то:

$$Target \rightarrow \langle Max, Any, Max \rangle,$$

что, впрочем, может быть недостижимо из-за наличия многоцелевой оптимизации (если только результативность и ресурсоэкономность не имеют независимую максимизацию за счет снижения скорости обслуживания очереди); для случая же, когда необходимо пользователю предоставить возможность самостоятельно выбирать эффективность интерфейса с Сервисом, будет:

$$Target \rightarrow \langle Certain, Certain, Certain \rangle,$$

т. е. значения показателей задаются явно (как вводом конкретных чисел, так и «ползунками» для указания их отношения друг между другом).

Эксперимент

Для проверки общей работоспособности Модели, обоснования ее необходимости (из-за нетривиальности зависимостей между структурой/параметрами интерфейса и его эффективностью), а также получения предварительных оценок, был создан программный прототип для проведения соответствующих экспериментов (далее – Прототип); его разработка велась в среде Microsoft Visual Studio Community 2019 на языке программирования C#.

Работа прототипа состоит из следующей последовательности шагов.

Шаг 1. Строится интерфейс путем указания его форм, элементов, их взаимосвязей и параметров. В качестве заготовок интерфейса может быть взята стандартная конфигурация или сгенерирована группа новых согласно заданным условиям (например, для определенного числа форм и элементов на них).

Шаг 2. Задаются инвариантные параметры интерфейса, определяющие функции корректировки и атомарной эффективности (например, для кнопок перехода).

Шаг 3. Создается Модель согласно построенному интерфейсу и его инвариантным параметрам.

Шаг 4. Производится вычисление общей эффективности интерфейса с выводом итогового и промежуточных результатов на консоль.

Эксперименты были упрощены путем указания всех инвариантных эффективностей, равных значению некоторой эталонной – $\{0.9999, 0.9999, 0.9999\}$ (т. е. близкой, но не равной, к идеальной). Сами же инварианты состояли из следующих эффективностей – для элементов всех классов, данных всех типов, формы и кнопки перехода. Учет количества элементов на форме производился путем умножения эффективности формы на эталонную столько раз, сколько было на ней элементов; учет размера данных элемента осуществлялся аналогичным образом, но количество умножений бралось равным разрядности размера (например, для разрядности 1000 равно 3); а для корректировки эффективностей количество умножений равнялось соответствующему индексу. Также считалось, что все элементы относятся к классу текстовых полей и предназначены для отображения 1 символа.

Во всех экспериментах элементы равномерно размещались на формах, в случае некратности количества первых (N) ко вторым (M) «лишние» элементы размещались по формам случайным образом. Например, 10 элементов по 3 формам могли разместиться следующим образом: 3, 4 и 3. Такое распределение можно увидеть в следующем отладочном выводе Прототипа (в формате JSON).

```
{
  'Name': 'Simple interface by N = 3 and M = 10',
  'Forms': [ {
    'Name': 'Form 1',
    'Button': 'Next 1->2',
    'Elements': [{
      'Name': 'Element 1.1',
      'Class': 'TextBox',
      'Data': '<Text, 1>',
    }, {
      'Name': 'Element 1.2',
      'Class': 'TextBox',
      'Data': '<Text, 1>',
    }, {
      'Name': 'Element 1.3',
      'Class': 'TextBox',
      'Data': '<Text, 1>',
    }
  ]
}, {
  'Name': 'Form 2',
  'Button': 'Next 2->3',
  'Elements': [{
    'Name': 'Element 2.1',
    'Class': 'TextBox',
    'Data': '<Text, 1>',
  }, {
    'Name': 'Element 2.2',
    'Class': 'TextBox',
    'Data': '<Text, 1>',
  }, {
    'Name': 'Element 2.3',
    'Class': 'TextBox',
    'Data': '<Text, 1>',
  }
}, {
  'Name': 'Form 3',
  'Button': 'Next 3->4',
  'Elements': [{
    'Name': 'Element 3.1',
    'Class': 'TextBox',
    'Data': '<Text, 1>',
  }, {
    'Name': 'Element 3.2',
    'Class': 'TextBox',
    'Data': '<Text, 1>',
  }, {
    'Name': 'Element 3.3',
    'Class': 'TextBox',
    'Data': '<Text, 1>',
  }
}
]
```

```
{
  'Name': 'Element 2.4',
  'Class': 'TextBox',
  'Data': '<Text, 1>',
}, {
  'Name': 'Form 3',
  'Button': 'Next 3->4',
  'Elements': [{
    'Name': 'Element 3.1',
    'Class': 'TextBox',
    'Data': '<Text, 1>',
  }, {
    'Name': 'Element 3.2',
    'Class': 'TextBox',
    'Data': '<Text, 1>',
  }, {
    'Name': 'Element 3.3',
    'Class': 'TextBox',
    'Data': '<Text, 1>',
  }
}, {
  'Name': 'Form 3',
  'Button': 'Next 3->4',
  'Elements': [{
    'Name': 'Element 3.1',
    'Class': 'TextBox',
    'Data': '<Text, 1>',
  }, {
    'Name': 'Element 3.2',
    'Class': 'TextBox',
    'Data': '<Text, 1>',
  }, {
    'Name': 'Element 3.3',
    'Class': 'TextBox',
    'Data': '<Text, 1>',
  }
}
]
```

Были проведены эксперименты по следующим сценариям построения интерфейсов.

Сценарий 1. Перебор количества форм интерфейса ($N = 1..100$) при неизменном количестве его элементов ($M = 100$). В рамках эксперимента было взято $M = 100$ элементов, которые равномерно размещались на формах, количество N которых менялось от 1 до 100.

Вычисление эффективности интерфейса такой конфигурации для различного количества форм N представлено в виде гистограммы ее показателя на рисунке 2а; показатель является любым из трех, поскольку, согласно условиям эксперимента, они равнозначны.

Гистограмма имеет возрастающее значение показателя эффективности с достижением пика 0.854697349 при количестве форм, равных 9 (отмечено красной линией (см. рисунок 2а) и соответствует наивысшей высоте столбца), с последующим его уменьшением. Таким образом, в сценарии существует некоторое оптимальное соотношение количества форм и элементов на них, при котором показатель эффективности будет максимальным.

Сценарий 2. Перебор количества элементов интерфейса ($M = 1..100$) при неизменном количестве его форм ($N = 100$). В рамках эксперимента было взято $N = 100$ форм, на которых равномерно размещались элементы, количество M которых менялось от 1 до 100.

Вычисление эффективностей интерфейса такой конфигурации для различного количества элементов M представлено в виде гистограммы ее показателя на рисунке 2б. В отличие от гистограммы на рисунке 2а, данная характеризуется уменьшением показателя эффективности при увеличении элементов. Максимальное значение показателя (отмечено красной линией) в эксперименте было достигнуто для 1 элемента и равнялось 0.5911850000251.

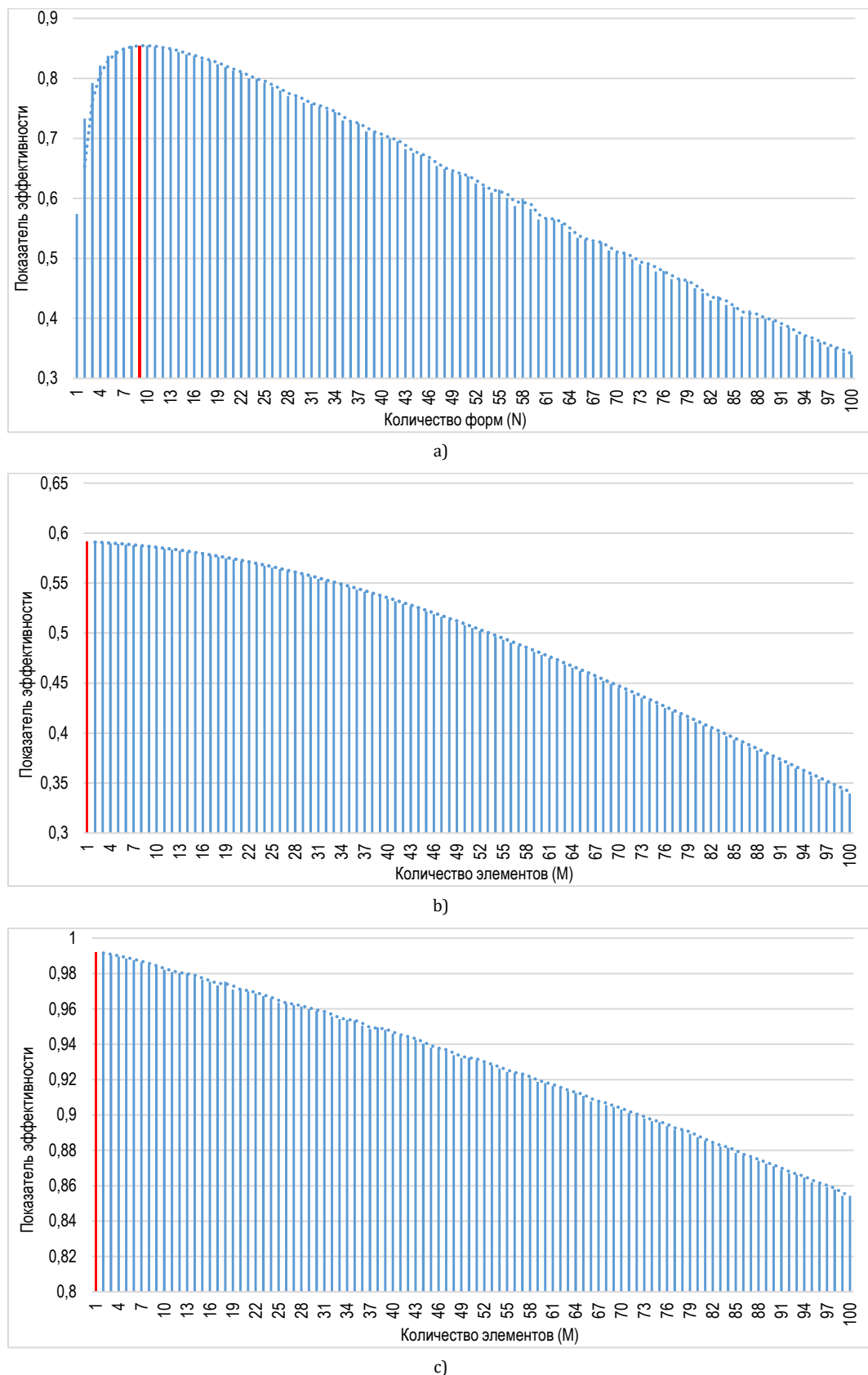


Рис. 2. Зависимость показателя эффективностей интерфейса для равномерного распределения 100 элементов по переменному количеству форм (а), переменного количества элементов по 100 формам (б) и по 10 формам (с)

Fig. 2. Dependence of the Interface Efficiency Indicator for Uniform Distribution of 100 Elements across a Variable Forms Number (a), a Variable Number Elements across 100 Forms (b) and 10 Forms (c)

Результат эксперимента по Сценарию 2 является закономерным, поскольку любое увеличение количества элементов (гипотетически, не зависимо от количества форм) ведет к уменьшению эффективности интерфейса; естественно, при решении задачи проектирования интерфейса данным параметром «играться» нельзя, поскольку он определяется неизменной логикой работы Сервиса.

Сценарий 3. Перебор количества элементов интерфейса ($M = 1..100$) при неизменном количестве его форм ($N = 10$). В рамках эксперимента было взято $N = 10$ форм, на которых равномерно размещались элементы, количество M которых менялось от 1 до 100.

Вычисление эффективностей интерфейса такой конфигурации для различного количества элементов M представлено в виде гистограммы ее показателя на рисунке 2с. Аналогично гистограмме на рисунке 2б, максимальное значение показателя эффективности (отмечено красной линией) в эксперименте было достигнуто для 1 элемента и равнялось 0.991932315. Результат аналогичен Сценарию 2 и доказывает, что уменьшение количества элементов более существенно, чем изменение количества форм.

Результаты экспериментов позволяют сделать следующие выводы. Во-первых, Модель достаточно точно отражает особенности интерфейса, поскольку получаемая эффективность существенно зависит от задаваемых параметров. Во-вторых, Модель не носит исключительно теоретический характер, поскольку может быть реализована с помощью программных средств. И, в-третьих, расчет эффективности интерфейса даже при условии сильных допущений является не тривиальной задачей, а ее максимизация требует проведения определенных вычислений или экспериментов.

Заключение

Работа посвящена задаче построения аналитической модели, позволяющей оценивать эффективность интерфейсов взаимодействия пользователя с информационным сервисом запросного типа –

т. е. иницируемым человеком и представляющим собой последовательность форм с вводимыми данными, на основании которых система выдает результат. В интересах этого, на основании типового примера такого взаимодействия выдвинуты 10 предпосылок к построению модели. Каждая из предпосылок переведена в формализованный вид, что позволило создать единую систему, отражающую взаимосвязь форм и элементов интерфейса, их структуры и параметров, а также являющуюся базисом для вычисления общей эффективности интерфейса.

Основным результатом исследования является эффективностная модель интерфейса информационного сервиса запросного типа.

Новизна результатов заключается в авторской модели оценки эффективности интерфейса информационной системы определенного типа, имеющей полностью формальный вид. Кроме того, впервые предложены 10 предпосылок к созданию моделей, которые могут быть адаптированы для достаточно большого класса интерфейсов.

Теоретическая значимость исследования заключается в расширении аппарата формализации и оценивания интерфейсной части информационного взаимодействия, изначально характеризующейся сравнительно высокой степенью субъективности.

Практическая значимость состоит в готовности эффективностной модели к программной реализации, что было продемонстрировано с помощью эксперимента по трем сценариям; полученные в результате зависимости показателей эффективности интерфейса от его структуры и параметров также несут определенную практическую ценность для архитекторов интерфейсов.

Продолжением исследования должна стать разработка полноценного средства моделирования интерфейса информационного сервиса запросного типа, а также проведение большего числа экспериментов.

Список источников

1. Израилов К.Е., Левшун Д.С., Чечулин А.А. Модель классификации уязвимостей интерфейсов транспортной инфраструктуры «умного города» // Системы управления, связи и безопасности. 2021. № 5. С. 199–223. DOI:10.24412/2410-9916-2021-5-199-223
2. Курта П.А. Взаимодействие пользователя с информационной системой. Часть 3. Оценка эффективности // Известия СПбГЭТУ ЛЭТИ. 2021. № 4. С. 58–72.
3. Самарин М.А., Максимов А.В., Вострых А.В. Повышение эффективности информационных систем типа электронные библиотеки // Природные и техногенные риски (физико-математические и прикладные аспекты). 2023. № 1 (45). С. 84–93.
4. Буйневич М.В., Курта П.А. Методология исследования метода оптимизации информационного взаимодействия в аспекте решения задачи пользователя // Информационные технологии и телекоммуникации. 2019. Т. 7. № 4. С. 50–58. DOI:10.31854/2307-1303-2019-7-4-50-58
5. Ломоносова Т.С. Информационный сервис важность информационных услуг в жизни человека // Теория и практика современной науки. 2015. № 2(2). С. 87–88.

6. Вострых А.В. Метод оценки эффективности графических пользовательских интерфейсов программных продуктов // Приборы и системы. Управление, контроль, диагностика. 2022. № 10. С. 19-28. DOI:10.25791/pribor.10.2022.1365
7. Ярошенко А.Ю. Предпосылки к необходимости непрерывного ранжирования требований пожарной безопасности // Национальная безопасность и стратегическое планирование. 2021. № 3(35). С. 100–105. DOI:10.37468/2307-1400-2021-3-100-105
8. Любчикова О.В. Типы интерфейсов и виды элементов // Новая наука: От идеи к результату. 2016. № 12-3. С. 121–124.
9. Трегубов А.С. Разработка адаптивных контекстозависимых интерфейсов с использованием онтологических моделей // Кибернетика и программирование. 2017. № 6. С. 50–56. DOI:10.25136/2306-4196.2017.6.24747
10. Izrailov K., Levshun D., Kotenko I., Chechulin A. Classification and analysis of vulnerabilities in mobile device infrastructure interfaces // Communications in Computer and Information Science. 2022. Vol. 1544. PP. 301–319. DOI:10.1007/978-981-16-9576-6_21
11. Антюхов В.И., Остудин Н.В., Ярошенко А.Ю., Черных А.К. Информационная потребность должностных лиц Центров управления в кризисных ситуациях (ЦУКС) МЧС России // VII Международная научно-практическая конференция «Сервис безопасности в России: опыт, проблемы, перспективы. Обеспечение безопасности при чрезвычайных ситуациях» (Санкт-Петербург, Российская Федерация, 24 сентября 2015). Санкт-Петербург: Санкт-Петербургский университет ГПС МЧС России, 2015. С. 70–71.
12. Грушко Ю.В. Аппаратно-программный комплекс аугментативной системы коммуникации на основе технологии Eyetracking // Вестник КРАУНЦ. Физико-математические науки. 2019. Т. 27. № 2. С. 55–73. DOI:10.26117/2079-6641-2019-27-2-55-73

References

1. Izrailov K.E., Levshun D.S., Chechulin A.A. Vulnerability Classification Model for Smart City Transport Infrastructure Interfaces. *Systems of Control, Communication and Security*. 2021;5:199–223. DOI:10.24412/2410-9916-2021-5-199-223
2. Kurta P.A. Interaction of the User with the Information System. Part 3. Efficiency Evaluation. *Proceedings of Saint Petersburg Electrotechnical University Journal*. 2021;4:58–72.
3. Samarin M.A., Maximov A.V., Vostrikh A.V. Improving the Efficiency of Information Systems Such as Electronic Libraries. *Natural and Man-Made Risks (Physico-Mathematical and Applied Aspects)*. 2023;1(45):84–93.
4. Buinevich M., Kurta P. Research Methodology Of The Method Of Optimization Of Information Interaction In The Aspect Of Solving The User's Problem. *Telecom IT*. 2019;7(4):50–58. DOI:10.31854/2307-1303-2019-7-4-50-58.
5. Lomonosova T.S. Information Service the Importance of Information Services in Human Life. *Teoriya i praktika sovremennoy nauki*. 2015;2(2):87–88.
6. Vostrykh A.V. A Method for Evaluating the Effectiveness of Graphical User Interfaces of Software Products. *Instruments and Systems: Monitoring, Control, and Diagnostics*. 2022;10:19–28. DOI:10.25791/pribor.10.2022.1365
7. Yaroshenko A. Background to the Necessity of Continuous Ranking of Fire Safety Requirements. *National Security and Strategic Planning*. 2021;3(35):100–105. DOI:10.37468/2307-1400-2021-3-100-105
8. Lyubchikova O.V. Development of adaptive context-sensitive interfaces using ontological models. *Novaya nauka: Ot idei k rezul'tatu*. 2016;12-3:121–124.
9. Tregubov A.S. Development of adaptive context-sensitive interfaces using ontological models. *Cybernetics and programming*. 2017;6:50–56. DOI:10.25136/2306-4196.2017.6.24747
10. Izrailov K., Levshun D., Kotenko I., Chechulin A. Classification and analysis of vulnerabilities in mobile device infrastructure interfaces. *Communications in Computer and Information Science*. 2022;1544:301–319. DOI:10.1007/978-981-16-9576-6_21
11. Antyukhov V.I., Ostudin N.V., Yaroshenko A.YU., Chernykh A.K. Information needs of officials of the Crisis Management Centers of the Ministry of Emergency Situations of Russia. Proceedings of the VII International Scientific and Practical Conference "Security Service in Russia: Experience, Problems, Prospects. Ensuring Safety in Emergency Situations", 24 September 2015, St. Petersburg, Russian Federation. St. Petersburg: Saint-Petersburg State Fire Service University of the Ministry of RF for Civil Defense, Emergencies and Elimination of Consequences of Natural Disasters Publ.; 2015. p.70–71.
12. Grushko Y.V. Hardware and Software Complex of Augmentation Communication System on the Basis of Eyetracking Technologys. *Bulletin KRASEC. Physical and Mathematical Sciences*. 2019;27(2):55–73. DOI:10.26117/2079-6641-2019-27-2-55-73

Статья поступила в редакцию 01.12.2023; одобрена после рецензирования 08.12.2023; принята к публикации 11.12.2023.

The article was submitted 01.12.2023; approved after reviewing 08.12.2023; accepted for publication 11.12.2023.

Информация об авторе:

КУРТА | технический директор ООО «Норд клининг»
Павел Андреевич |  <https://orcid.org/0009-0005-6073-8626>

СПбГУТ)))

Санкт-Петербургский государственный университет
телекоммуникаций им. проф. М. А. Бонч-Бруевича

13TH INTERNATIONAL CONFERENCE ON ADVANCED INFOTELECOMMUNICATIONS ICAIT 2024
Международная научно-техническая и научно-методическая конференция
«Актуальные проблемы инфотелекоммуникаций в науке и образовании»



АПИНО
ICAIT

ОСНОВНЫЕ НАУЧНЫЕ НАПРАВЛЕНИЯ:

-  Радиотехнологии связи
-  Инфокоммуникационные сети и системы
-  Информационные системы и технологии
-  Теоретические основы радиоэлектроники и систем связи
-  Цифровая экономика, управление и бизнес-информатика
-  Гуманитарные проблемы информационного пространства
-  Сети связи специального назначения

27-28
ФЕВРАЛЯ 2024

ПОДРОБНОСТИ НА САЙТЕ КОНФЕРЕНЦИИ

APINO.SPBGUT.RU



22
31
десятилетие
науки и технологий

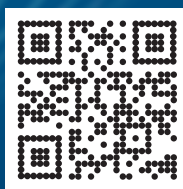
23—26 апреля 2024

СВЯЗЬ

36-я международная
выставка «Информационные
и коммуникационные технологии»

Экспозиция «Навитех» —
«Навигационные системы, технологии и услуги»

www.sviaz-expo.ru



Россия, Москва,
ЦВК «ЭКСПОЦЕНТР»



12+
Реклама

65
ЭКСПОЦЕНТР

Организатор

При поддержке

МИНПРОМТОРГ
РОССИИ

Под патронатом



В рамках

Российская неделя
высоких технологий
РНВТ

Выходные данные



Товарный знак №929373, правообладатель: Федеральное государственное бюджетное образовательное учреждение высшего образования «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича», 191186, Санкт-Петербург, наб. реки Мойки, 61, литера А (RU)
Зарегистрирован в Государственном реестре товарных знаков и знаков обслуживания Российской Федерации 13.03.2023 г. Заявка №2022733914

План издания научной литературы 2023 г., доп. п. 1

Дата выхода в свет	Усл.-печ. л.	Формат	Тираж	Заказ	Свободная цена
29.12.2023	16	60×84 _{1/8}	1000 экз.	№ 1561	

Ответственный редактор **Татарникова И.М.**

Выпускающий редактор **Яшугин Д.Н.**

Дизайн: **Коровин В.М.**

Обложка: <https://satellitemap.space/?constellation=starlink>

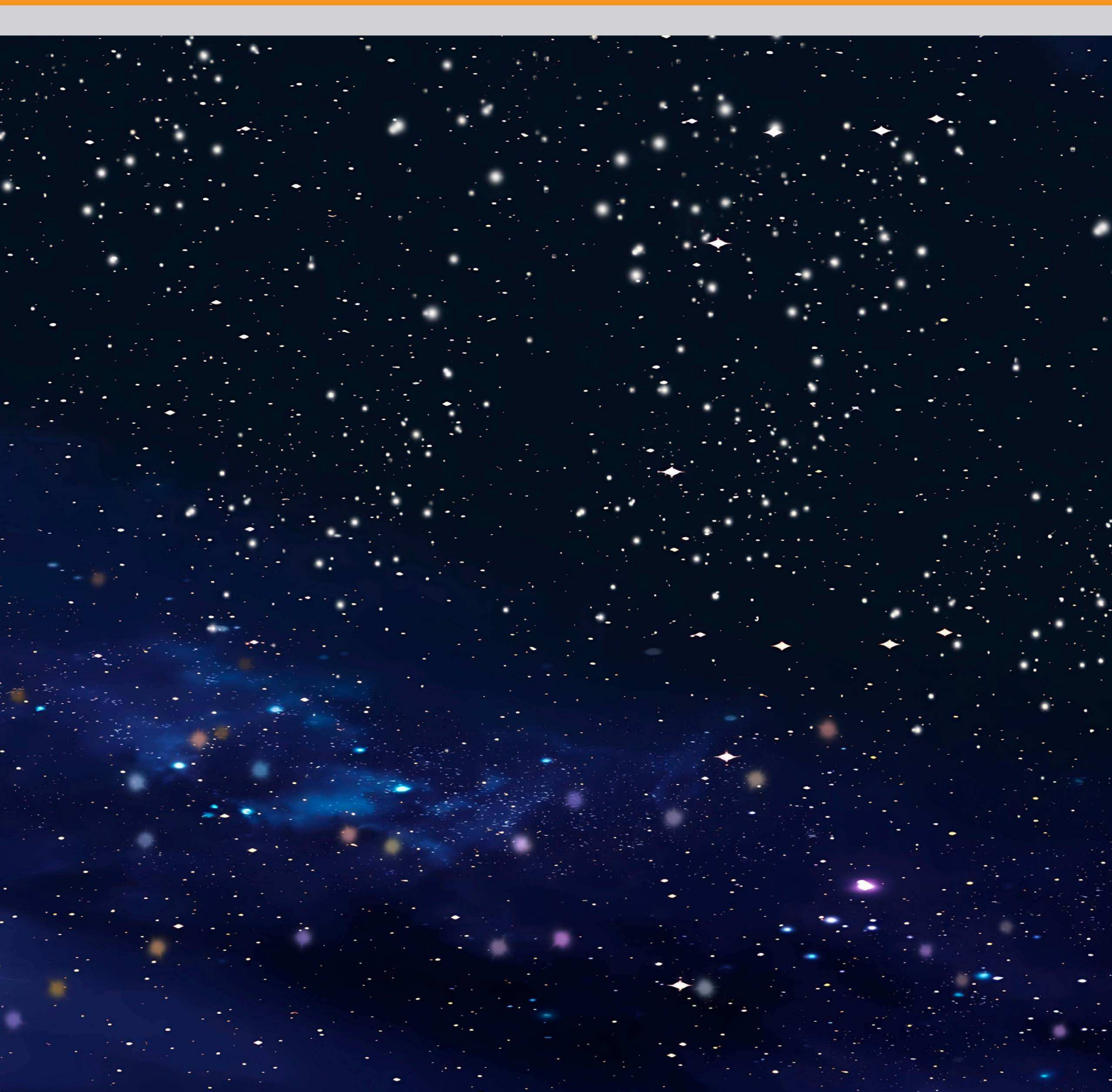
Адрес СПбГУТ:

19323, Санкт-Петербург, пр. Большевиков, 22/1

Учредитель и издатель:

Федеральное государственное бюджетное образовательное учреждение
высшего образования "Санкт-Петербургский государственный университет
телекоммуникаций им. проф. М.А. Бонч-Бруевича"

E-mail: tuzs@sut.ru Web: tuzs.sut.ru VK: vk.com/spbtuzs



Подписной индекс в Объединенном каталоге "ПРЕССА РОССИИ" - 59983